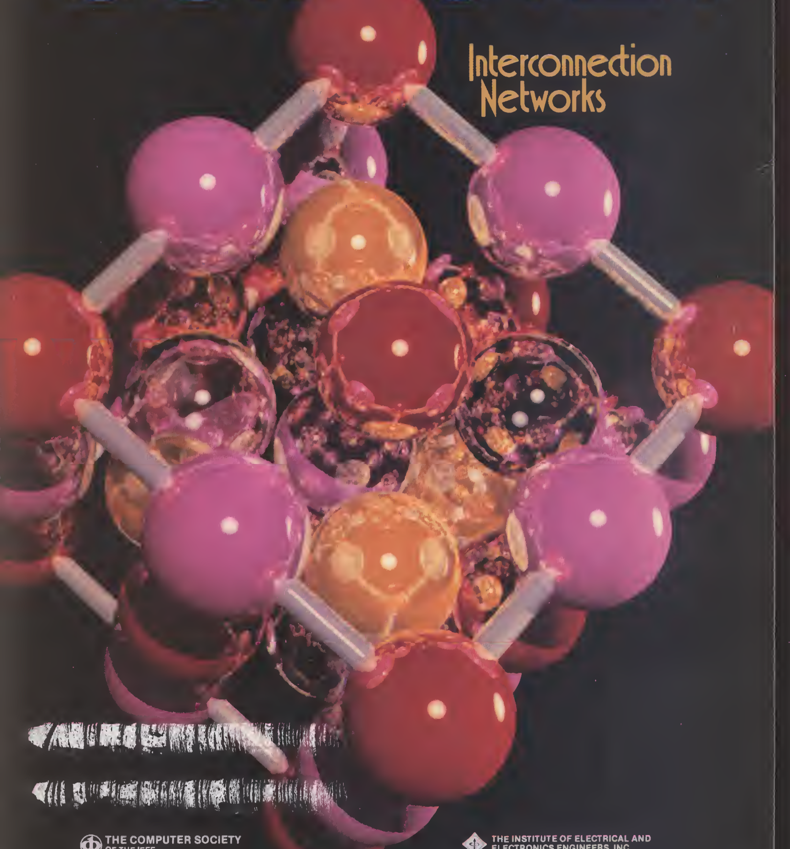
JUNE 1987

COMPUTER



Interconnection Networks

[redacted]

THE COMPUTER SOCIETY
OF THE IEEE

THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.

ENGINEERS—SCIENTISTS

Define Tomorrow at BDM.

If you value professional challenge and want to grow, join a company that's shaping the future: BDM. We're one of America's foremost technical and professional services firms with a sustained growth rate of 30% per year. And we're committed to strengthening the nation in meaningful ways through our efforts in defense, communications, space, and other vital areas.

Using state-of-the-art technologies, you'll take your capabilities directly to our clients, working to create effective new solutions to difficult problems. You'll work in an environment that's fast-paced and innovative—designed especially with your entrepreneurial spirit in mind.

Exciting technical challenges exist now in the following BDM locations:

McLean, VA
Albuquerque, NM
Los Angeles, CA
Columbia, MD
Dayton, OH
Seattle, WA
Tacoma, WA
Huntsville, AL
Austin, TX

We're looking for skilled professionals, preferably with an MS or PhD degree, with experience in the following areas:

ARTIFICIAL INTELLIGENCE
- Aerospace Computing
  - —Expert Systems Applications
  - —Aerospace Manufacturing Technology
- Computing Technology
  - —Parallel Computer Structures
  - —Distributed Computing/Operating Systems
  - —Parallel and Multiprocessor Computer Systems, Architecture, and Distributed Operating Systems
- Knowledge-Based Systems
  - —Planning and Reasoning Methods
  - —Knowledge Elicitation/Acquisition
  - —Representation and Design
  - —AI Software Tools/Workstations
  - —Natural Language, Vision, Speech
- Optical Computing Technology
  - —Optical Computing and Neural Analogue Systems
  - —Algotecture Analysis

SOFTWARE ENGINEERING
- SW Test & Evaluation
- SW IV&V
- Optical Disk Storage Technology
- Computer Security
- Ada, Pascal, Fortran, Lisp
- UNIX, 'C'
- Ethernet, Decnet
- Zenith Micros
- Embedded SW T&E
- Decision Support
- SW Graphics
- Sun Workstations/Internetworks
- Computer Integrated Manufacturing

DATABASE MANAGEMENT SYSTEMS
- ORACLE
- IDMS/R
- M204
- DATACOM DB, ADR Products
- INGRES
- Database Security
- Relational Database Design

MANAGEMENT INFORMATION SYSTEMS
- COBOL, ISPF
- IBM Mainframe
- Large-Scale MIS Development
- Project Leadership

If you have experience in any of these areas, we would like to talk to you. Please send your resume and geographical preferences in confidence to: BDM International, Inc., Holly K. Marcario, Dept. CM687, 7915 Jones Branch Drive, McLean, VA 22102. An Equal Opportunity Employer. U.S. Citizenship Required.

BDM

Announcing Simulation Conference 11
August 31 - September 3, 1987, San Diego

Four simulation sessions -- register now for one or all

- **Military Applications**
- **Simulation Methods**
- **Network Analysis**
- **Factory Planning**

New simulation tools, techniques, and off-the-shelf-models that help you succeed

This is your once-a-year opportunity to meet with other developers and users of simulation. Modern approaches to simulation will be discussed and demonstrated.

- You see how simulation results can now be presented through animation and other easy-to-understand graphics.
- You learn how new tools and techniques help you greatly in overcoming problems of model design, documentation, implementation, validation, and evolution.
- You learn about general models in major application areas. You simply enter parameters describing your scenario. The simulation phase follows immediately--no programming delays and costs.

Register now for one or more of the following four simulation sessions.

Military Applications
August 31

Modelling issues such as artificial intelligence, SDI, simulation based training. Status and plans for major wargaming and logistics models. Tutorial and demonstration of the effectiveness of animated results from wargaming models. Tutorial demonstration of SIMTRAINER™, a new simulation with animation based training system.

Simulation Methods
September 1

Simulation modelling issues such as statistical considerations, artificial intelligence, combined discrete-continuous, teaching simulation, and simulation on parallel processors. A tutorial and demonstration of SIMSCRIPT II.5® with animation.

Network Analysis
September 2

Network issues such as tradeoffs and performance prediction. Major applications areas such as computer and communications, military, telecommunications, and manufacturing. A tutorial and demonstration of new NETWORK II.5® with animation.

Factory Planning
September 3

Factory planning issues such as throughput and bottleneck analysis, layout planning, equipment and personel utilization. Major factory planning application areas: production scheduling, inventory analysis, capacity planning and material handling. A tutorial and demonstration of a complete factory planning session using SIMFACTORY® with animation.

Conference Hotel

Holiday Inn Embarcadero--a short ride from the San Diego airport. Call (619) 232-3861 and ask for the special rate.

Registration is limited -- call now

For immediate registration, call:
Art Hamilton at (619) 457-9681
CACI, 3344 N. Torrey Pines Ct.,
La Jolla, California 92037.

Register now to save 50%
Space is limited

This important conference is limited to 500 participants, and San Diego hotel rooms are scarce.

Sign us up for:	**Reduced Price**
☐ Military Applications August 31	$50.00
☐ Simulation Methods September 1	$50.00
☐ Network Analysis September 2	$50.00
☐ Factory Planning September 3	$50.00

Name ______________________

Organization ______________________

Address ______________________

City __________ State ______ Zip ______

Telephone ______________________

☐ Enclosed is a check for $________.
☐ Bill my organization.
☐ Reserve ___________ places while we seek approval.

Return this form to: IEEE COMP

Art Hamilton
Conference Registrar
CACI
3344 North Torrey Pines Court
La Jolla, California 92037
Telephone (619) 457-9681

COMPUTER

June 1987 **Published by the Computer Society of the IEEE** Vol. 20, No. 6

ARTICLES

9 **Guest Editor's Introduction: Interconnection Networks for Parallel and Distributed Proccesing**
Laxmi N. Bhuyan

14 **A Survey and Comparison of Fault-Tolerant Multistage Interconnection Networks**
George B. Adams III, Dharma P. Agrawal, and *Howard Jay Seigel*
Fault model and fault-tolerance criterion are essential metrics for any comparison of network fault-tolerance capabilities.

30 **Augmented Shuffle-Exchange Multistage Interconnection Networks**
V.P. Kumar and *S.M. Reddy*
A simple yet effective technique reduces the problems of poor performance and low reliability associated with unique path multistage interconnection networks.

42 **Multiple Bus Architectures**
T.N. Mudge, J.P. Hayes, and *D.C. Winsor*
By replacing the single shared bus of conventional multiprocessor architectures by a set of buses, the number of processors can be increased to a few hundred.

50 **Optical Crossbar Networks**
A.A. Sawchuk, B.K. Jenkins, C.S. Raghavendra, and *A. Varma*
Optical techniques solve fundamental problems in data communications. This article describes several design concepts for optical matrix-vector implementations of dynamic crossbar networks employing bulk (3-D) optical information processing architectures.

63 **The Performance of Multicomputer Interconnection Networks**
Daniel A. Reed and *Dirk C. Grunwald*
You cannot find an interconnection network optimal for *all* applications but, by employing the methods described here, you can choose one that fits the requirements of certain classes of applications well.

Circulation: *Computer* (ISSN 0018-9162) is published monthly by the Computer Society of the IEEE. IEEE Headquarters: 345 East 47th St., New York, NY 10017; Computer Society of the IEEE, Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 821-8380. Annual subscription included in society member dues. Nonmember prices: available upon request. Single copy prices: members $7.50; nonmembers $15.00. This magazine is also available in microfiche form.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854. Second class postage is paid at New York, New York, and at additional mailing offices.

Copyright and reprint permissions: Copyright © 1987 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons: (1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 29 Congress Street, Salem, MA 01970. (2) pre-1978 articles without fee. Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. For other copying, reprint, or republication permission, write to True Seaborn, Editor and Publisher, *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720. All rights reserved. Copyright © 1987 by the Institute of Electrical and Electronics Engineers, Inc.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computer* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space considerations. *Computer* subscribes to the Computer Press Association's code of professional ethics.

DEPARTMENTS

76 Open Channel
My personal profiles of programmer personnel; Look before you leap to buy new technology

78 Standards
Proposed design automation standards

82 Update

84 Computer Society News
CS nominees announced; CS auditor's report; Wang Institute support urged

89 Conferences

92 Calendar

97 Call for Papers

100 New Products

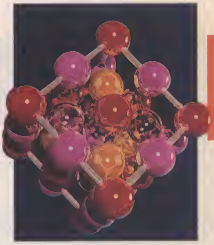
108 New Literature

109 Microsystem Announcements

110 IC Announcements

115 Book Reviews

Career Opportunities 111, Advertiser/Product Index 120, Reader Service Card 120A, Change-of-Address Form 91, Membership Application 88D



On the cover

Computer graphic depiction of a metal alloy generated by Melvin L. Prueitt, Motion Picture/TV Production, Los Alamos National Laboratory. Data supplied by Fred Mueller, Center for Material Science, Los Alamos National Laboratory.

Cover design by Jay Simpson

In the next issue

Systolic arrays

COMPUTER
10662 Los Vaqueros Circle, Los Alamitos, CA 90720

Editor-in-Chief: Bruce D. Shriver, IBM

Editorial Board
Dharma P. Agrawal, North Carolina State University
Scott Davidson, Western Electric
Richard Eckhouse, Moco Inc.
Gideon Frieder, University of Michigan
Edmund Gallizzi, Eckerd College
John Gurd, University of Manchester
James H. Haynes, University of California, Santa Cruz
Ronald G. Hoelzeman, University of Pittsburgh
Franklin F. Kuo, SRI International
Wiley McKinzie, Rochester Institute of Technology
K. M. Mohiuddin, IBM Almaden Research Lab.
William Riddle, Software Productivity Consortium
Sallie Sheppard, Texas A&M University
Harry Strong, Mitre Corporation
Helen M. Wood, National Bureau of Standards

Submissions: Send six double-spaced copies of articles and special-issue proposals to Bruce D. Shriver, IBM T.J. Watson Research Center, PO Box 704, H0-B04A, Yorktown Heights, NY 10598; (914) 789-7626. For electronic submission, Shriver's addresses are Compmail+, b.shriver; CSnet, Shriver@ibm.com; Vnet, shriver at yktvmh.

Staff
Editor and Publisher: True Seaborn
Managing Editor: Marilyn Potes
Issue Editor: Nancy Hays
Assistant Editors: Louise Anderson, Tom Szalkiewicz
Contributing Editor: Ware Myers
Assistant to the Publisher: Pat Paulsen
Art Director: Jay Simpson
Design/Production: Miriam Wiegel
Production Supervisor: David Gaines
Circulation Manager: Christina Champion
Advertising Director: Dawn Peck
Advertising Coordinators: Heidi Rex, Marian Tibayan

Magazine Advisory Committee
Michael Evangelist (chair), Vishwani D. Agrawal, James J. Farrell III, Ted Lewis, David Pessel, True Seaborn, Bruce D. Shriver, John Staudhammer

Publications Board
J. T. Cain (chair), Vishwani D. Agrawal, J. Richard Burke, Gerald L. Engel, Michael Evangelist, James J. Farrell III, Lansing Hatfield, Ronald G. Hoelzeman, Ted Lewis, Ming T. Liu, Ez Nahouraii, David Pessel, C.V. Ramamoorthy, Vincent Shen, Bruce D. Shriver, John Staudhammer, Steven L. Tanimoto

IEEE INTERNATIONAL CONFERENCE ON COMPUTER DESIGN: VLSI IN COMPUTERS

Sponsored by: **The Computer Society of the IEEE and IEEE Circuits and Systems Society**
In cooperation with: **The IEEE Electron Device Society**

ICCD '87

Rye Town Hilton, Rye Brook, New York
October 5-October 8, 1987

THE INTERDISCIPLINARY CONFERENCE

- ☐ VLSI and Technology
- ☐ Design and Test
- ☐ Computer-Aided Design
- ☐ Architectures and Algorithms

The focus of this multi-discipline conference is intended to emphasize the interactions between system design, logic circuit design, memory design, architecture, software, CAD, Testing, Physical Design, and VLSI Technology. The objectives of this conference will offer to those attending an alternative view into the VLSI CAD environment. Unlike other conferences focusing on the generalized aspects of DA, ICCD stresses the interrelationships which exist in the CAD Industry.

HIGHLIGHTS

41 SESSIONS ■ Algorithms ■ Object-Oriented Techniques for CAD ■ Synthesis and Floorplanning for VLSI ■ Methods in Test Pattern Generation ■ Multiprocessors ■ Placement Techniques on Parallel Architectures ■ Local Area Network Controllers ■ System Packaging ■ Computer Design ■ Synthesis of DSP Systems at Leuven ■ DEC 32-Bit Processor Family ■ National Semiconductors NS32532 Microprocessor ■ Symbolic Layout Compaction Benchmarks ■ Self-Timed Design ■ VLSI Technologies ■ Special-Purpose Architectures ■ Routing ■ Knowledge-Based Approaches for Design and Test ■ Reliable Systems ■ Graphics and Image Processing ■ New Approaches in VLSI Design ■ Storage Technology for the 90's ■ Digital Signal Processing ■ Integrated CAD Systems ■ Design of Arithmetic Components ■ Physical Verification ■ Built-In-Self-Test in Practice ■ State-of-the-Art Technology Integration ■ Fault-Tolerant Computing ■ Symbolics Ivory Lisp Processor ■ Performance-Oriented Design ■ CCI Regulus Supermini ■ High-Level Synthesis ■ Design of MC8030 Microprocessor ■ High-Level Language Architectures ■ Advances in Physical Design ■ New Methods in Design ■ Architecture of Arithmetic Units ■ Logic Simulation ■ PLA Design and Testing

PLENARY SESSIONS

- **Design & Test**
  Semiconductor and Silicon Compilers effect on Design and Test; Trends and Directions
- **CAD**
  Computer Aided Fabrication of Integrated Circuits
- **VLSI & Technology**
  The Development of High TC Superconductors and Their Application Potential
- **Architectures**
  TBA

THE COMPUTER SOCIETY
OF THE IEEE

THE INSTITUTE OF ELECTRICAL
AND ELECTRONICS ENGINEERS, INC
IEEE

ICCD '87

AIRLINE INFORMATION

An arrangement with the Computer Society of the IEEE's International Conference on Computer Design and United Airlines offers you the services of its toll-free reservations desk along with the compliments of discounts.

> 5% on any fare for which you qualify including First Class or maxi-saver. The discount can range from 40% to 70% off normal coach fares.

Those not qualifying for the above discount, may be qualified for a minimum of 40% off the coach fare with no minimum stay or advance purchase required.

To make a reservation for one of the above or for any promotional fare, simply follow these easy steps:

1) Call United's toll free number: 800-521-4041 7 days a week, 8am-11pm EST.
2) Give the IEEE-CS Account #: 7146C
3) United will arrange to mail tickets to your home or office, or you may purchase them from your local Travel Agent.*

*If you purchase from your local Travel Agent be sure you or your travel agent calls the United's Convention desk to make your arrangement. The special IEEE-CS fare is available only for the ICCD '87 Conference thru United's Convention desk.

NOTE: Mileage plus will receive full credit.

HOTEL RESERVATION FORM

IEEE INTERNATIONAL CONFERENCE ON COMPUTER DESIGN—ICCD '87

Name

Street

City/State/Zip

☐ Single room
☐ Double room—Additional occupant
☐ Charges on one bill
☐ Bill separately

Arrival Date & Time Departure Date

Credit Card Name

Credit Card No Expiration Date

Cardholder's Name

Signature

Return To: Front Office Manager, ICCD '87
The Rye Town Hilton
699 Westchester Avenue
Rye Brook, New York 10573
Phone: (914) 939-6300

To guarantee your reservation, either a check for one night's lodging, or appropriate credit card information, must be included with the hotel reservation form.

Notes: Please indicate accommodations desired. Rooms should be claimed by 6 pm. We will attempt to accommodate you first at the Rye Town Hilton (914-939-6300). (Single $102/Double $120). It is your responsibility to find your own roommate. Reservations must be received no later than September 14, 1987. After that date reservations must be made directly with hotels on an availability basis. If reservations are made by telephone you must identify ICCD '87.

REGISTRATION FORM ICCD '87

	IEEE MEMBER	NON-MEMBER	STUDENT	AMOUNT
REGISTRATION				
Until September 18	$160	$200	$65 (w/o Proc.)	
After September 18	$210	$265	$65 (w/o Proc.)	
BANQUET	$32	$32	$32	
LUNCHEON	$14 per day	$14 per day	$14 per day	
☐ October 5, 1987	☐ October 6, 1987	☐ October 7, 1987		
			TOTAL	

IEEE or IEEE Computer Society Membership No.

Name Organization

Street Address

City/State/Zip

Credit Card Name Credit Card No. Expiration Date

Cardholder's Name Signature

Return To: Ms. Glenda D. McBride, Registrar, ICCD '87
Computer Society of the IEEE
1730 Massachusetts Ave., N.W.
Washington, D.C. 20036-1903
Phone: (202) 371-1013

To guarantee your reservation, either a check or appropriate credit card information (Visa, MasterCard, or American Express), must be included with the conference registration form.

Notes: Conference registration fee includes admission to the technical sessions, one copy of the conference proceedings (except students), break refreshments, and reception. Luncheons and banquet spaces are guaranteed only for advanced registrants. Otherwise, meals, literature, and other activities may be purchased at the conference on an as available basis. Refunds (less $15 handling fee) will be made if request is received in writing by October 2, 1987. After that date no refunds can be made.

The Eleventh Annual
International Computer
Software & Applications Conference
compsac 87

Takanawa Prince Hotel, Tokyo, Japan

Tutorials: October 5-6, 1987
Conference: October 7-9, 1987
Technical Tours: October 12-14, 1987

11

A Software Industry State-of-the-Art International Conference

Four Pre-conference Tutorials on Important Subjects (October 5-6, 1987)

- **Software Quality Assurance: A Practical Approach** by Tsun S. Chow, AT&T Bell Laboratories
- **Security of Integrated Engineering Information** by Robert E. Fulton, Georgia Inst. of Technology
- **Major Technology Impacts on Software Engineering** by Morio Nagata, Keio Univ.; Toshiro Ohno, Japan BAC; Koichi Furukawa, ICOT; Akira Nagashima, IPA; and Ken Sakamura, Univ. of Tokyo
- **Managing Software Projects** by Donald J. Reifer, Reifer Associates

Each tutorial runs from 9 a.m. to 5 p.m.

Conference Topic Areas
October 7-9, 1987

Each day starts with a plenary address or panel discussion by worldwide leaders in computer science, technology, and industry.

Four three-day conference tracks on:

- Software Design and Management
- Software Quality and Maintenance
- Databases and Distributed Processing
- Language Processing, Expert Systems, and Protocols

in 36 technical and management sessions.

Post-Conference Technical Tours • (October 12-14, 1987)

- **Toshiba** Corporation in Kawasaki city, Kanagawa
- **IPA** (Information-technology Promotion Agency) in Akihabara, Tokyo
- **NEC** Corporation in Abiko city, Chiba
- **Electrotechnical Laboratory** in the Tsukuba scientific city, Ibaraki
- **Hitachi** Corporation in Hatano city, Kanagawa
- **Nissan** Motor Company in Zama, Kanagawa
- **NTT** Corporation in Yokosuka city, Kanagawa
- **Mitsubishi** Electric Corporation in Kamakura city, Kanagawa
- **Fujitsu** Ltd. in Numazu city, Shizuoka
- **Matsushita** Industrial Company, Ltd. in Moriguchi city, Osaka
- **Kyoto University,** Dept. of Information Science in Kyoto city

These are complete tours with bus service to and from the Takanawa Prince Hotel each day and lunch with engineers and scientists of the companies visited.

Individuals living outside Asia, Australia & New Zealand— For additional information and registration, contact:
David H. Jacobsohn
COMPSAC 87 Registration Co-Chair
5642 South Harper Avenue
Chicago, IL 60637, USA
Telephone: 312/752-4562

Individuals living outside Asia, Australia & New Zealand— For travel, hotel and additional tour arrangements with major discount rates, contact the following company ASAP:
Commerce Tours International, Inc.
870 Market Street, Suite 920
San Francisco, CA 94102, USA
Telephone: 415/433-3072
Fax: 415/788-3280
Telex: 278124 CTI UR

Individuals within Asia, Australia & New Zealand, contact:
Secretariat
COMPSAC 87
c/o Business Center for Academic Societies Japan
Yamazaki Bldg 4F.
2-40-14, Hongo, Bunkyo-ku
Tokyo 113, Japan
Phone: 03-817-5831
Fax: 03-817-5836
Telex: 02722288 BCJSP J

All inquiries & reservations for technical tours should be directed to:
Secretariat
COMPSAC 87
c/o Business Center for Academic Societies Japan
Yamazaki Bldg. 4F.
2-40-14, Hongo, Bunkyo-ku
Tokyo 113, Japan
Phone: 03-817-5831
Fax: 03-817-5836
Telex: 02722268 BCJSP J

THE COMPUTER SOCIETY OF THE IEEE

INFORMATION PROCESSING SOCIETY OF JAPAN

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC
IEEE

THE COMPUTER SOCIETY

A member society of the Institute of Electrical and Electronics Engineers, Inc.

Executive Committee

President: Roy L. Russo*
IBM T.J. Watson Research Center
PO Box 218, Route 134
Yorktown Heights, NY 10598
(914) 945-3085

President-Elect: Edward A. Parrish, Jr.*

Vice Presidents
Education: Michael C. Mulder (1st VP)*
Technical Activities: Kenneth R. Anderson (2nd VP)*
Area Activities: Willis K. King†
Conferences and Tutorials: James H. Aylor†
Membership and Information: Merlin G. Smith†
Publications: J.T. Cain†
Standards: Helen M. Wood

Secretary: Duncan H. Lawrie
Treasurer: Joseph E. Urban
Past President: Martha Sloan*
Division V Director: Martha Sloan
Division VIII Director: H. Troy Nagle†
Executive Director: T. Michael Elliott†

*voting member of the Board of Governors
†non-voting member of the Board of Governors

Board of Governors

Term Ending 1987	Term Ending 1988
Barry W. Boehm	Mario R. Barbacci
Paul L. Borrill	Victor R. Basili
Glen G. Langdon, Jr.	Lorraine M. Duvall
Duncan H. Lawrie	Michael Evangelist
Susan L. Rosenbaum	Allen L. Hankinson
Bruce D. Shriver	Laurel Kaleda
Harold S. Stone	Ted Lewis
Wing N. Toy	Ming T. Liu
Helen M. Wood	Earl E. Swartzlander, Jr.
Akihiko Yamada	Joseph E. Urban
Oscar N. Garcia†	

Next Board Meeting

8:30 a.m.-5 p.m., June 19, 1987,
Chicago Marriott Downtown

Senior Staff

Executive Director: T. Michael Elliott
Editor and Publisher: True Seaborn
Director, Computer Society Press: Chip G. Stockton
Director, Conferences: William R. Habingreither
Director, Finance and Administration: Mary Ellen Curto

Offices of the Computer Society

Headquarters Office
1730 Massachusetts Ave. NW
Washington, DC 20036-1903
General Information: (202) 371-0101
Publications Orders: (800) 272-6657
Telex: 7108250437 IEEE COMPSO

Publications Office
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
Membership and General Information: (714) 821-8380

European Office
2 Avenue de la Tanche
B1160 Brussels, Belgium
Phone: 32 (2) 660-11-43
Telex: 25387 AVVAL B

Purpose

The Computer Society strives to advance the theory and practice of computer science and engineering. It promotes the exchange of technical information among its 90,000 members around the world, and provides a wide range of services which are available to both members and non-members.

Membership

Members receive the highly acclaimed monthly magazine *Computer,* discounts on all society publications, discounts to attend conferences, and opportunities to serve in various capacities. Membership is open to members, associate members, and student members of the IEEE, and to non-IEEE members who qualify as affiliate members of the Computer Society.

Publications

Periodicals. The society publishes six magazines (*Computer, IEEE Expert, IEEE Computer Graphics and Applications, IEEE Design & Test of Computers, IEEE Micro, IEEE Software*) and three research publications (*IEEE Transactions on Computers, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Software Engineering*).

Conference Proceedings, Tutorial Texts, Standards Documents. The society publishes more than 100 new titles every year.

Computer. Received by all society members, *Computer* is an authoritative, easy-to-read monthly magazine containing tutorial, survey, and in-depth technical articles across the breath of the computer field. Departments contain general and Computer Society news, conference coverage and calendar, interviews, new product and book reviews, etc.

All publications are available to members, nonmembers, libraries, and organizations.

Activities

Chapters. Over 100 regular and over 100 student chapters around the world provide the opportunity to interact with local colleagues, hear experts discuss technical issues, and serve the local professional community.

Technical Committees. Over 30 TCs provide the opportunity to interact with peers in technical specialty areas, receive newsletters, conduct conferences, tutorials, etc.

Standards Working Groups. Draft standards are written by over 60 SWGs in all areas of computer technology; after approval via vote, they become IEEE standards used throughout the industrial world.

Conferences/Educational Activities. The society holds about 100 conferences each year around the world and sponsors many educational activities, including computing sciences accreditation.

European Office

This office processes Computer Society membership applications and handles publication orders. Payments are accepted by cheques in Belgian francs, pounds sterling, German marks, Swiss francs, or US dollars, or by American Express, Eurocard, MasterCard, or Visa credit cards.

Ombudsman

Members experiencing problems — late magazines, membership status problems, no answer to complaints — may write to the ombudsman at the Publications Office.

Information

Use the Reader Service Card to obtain the following material:

- Membership information and application (RS #202)
- Publications catalog (proceedings, tutorials, standards) (RS #201)
- Periodicals subscription application/information for individuals (members, sister-society members, others) (RS #200)
- Periodicals subscription application/information for organizations (libraries, companies, etc.) (RS #199)
- List of awards and award nomination forms (RS #198)
- Technical committee list and membership application (RS #197)
- Directory of officers, board members, committee chairs, representatives, staff, chapters, standards working groups, etc. (RS #196)

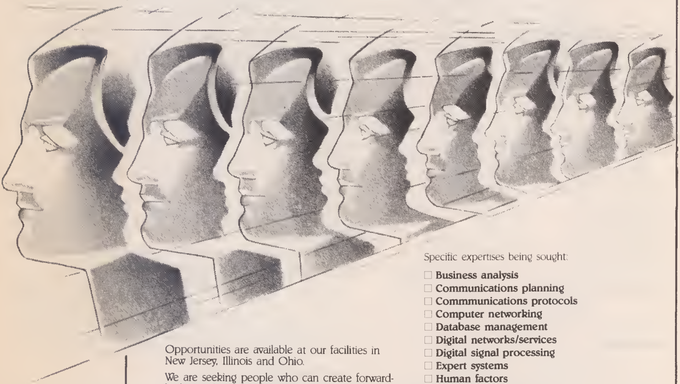
Also see membership application in this magazine.

Systems Engineers Systems Architects
Software Developers

AT&T Bell Laboratories

Where ideas flow

At AT&T Bell Laboratories, innovative people from a wide variety of technical backgrounds share their insights and knowledge to create forward-looking information systems, services and products. In this environment, every person has the opportunity to make unique contributions.



Opportunities are available at our facilities in New Jersey, Illinois and Ohio.

We are seeking people who can create forward-looking solutions, evaluate alternative architectures, perform competitive analyses, assess service viability, define, implement and measure systems or develop strategies to get products to market.

AT&T
The right choice.

If you have a Master's or PhD in Systems Engineering, Computer Science, Electrical Engineering, Mathematics, Physics, Statistics, or Operations Research with hardware and/ or software experience, or a BS in these fields with a minimum of 5 years highly relevant experience, please send your resume to:

Director, Technical Employment,
AT&T Bell Laboratories, Dept. 123/6923/87,
101 J.F. Kennedy Parkway, Short Hills,
NJ 07078-0905.

Specific expertises being sought:

- ☐ **Business analysis**
- ☐ **Communications planning**
- ☐ **Commmunications protocols**
- ☐ **Computer networking**
- ☐ **Database management**
- ☐ **Digital networks/services**
- ☐ **Digital signal processing**
- ☐ **Expert systems**
- ☐ **Human factors**
- ☐ **International applications**
- ☐ **Integrated networks**
- ☐ **Lightwave applications**
- ☐ **Local area networks**
- ☐ **Market definition for new technologies and services**
- ☐ **Measurement systems**
- ☐ **Network optimization tools**
- ☐ **Network planning and design**
- ☐ **Operations automation**
- ☐ **Packet switching**
- ☐ **Performance measurement and analysis**
- ☐ **Product planning**
- ☐ **Real time systems software**
- ☐ **Software development tools**
- ☐ **Subjective testing**
- ☐ **Systems architecture**
- ☐ **Systems planning**
- ☐ **Transmission systems**
- ☐ **UNIX™ systems applications**

An equal opportunity employer.

Guest Editor's Introduction

Interconnection Networks for Parallel and Distributed Processing

Laxmi N. Bhuyan
University of Southwestern Louisiana

With the rapid advances in technology, it is now feasible to build a system consisting of hundreds or thousands of processors. Processors in such a parallel/distributed system may spend a considerable amount of time just communicating among themselves unless an efficient interconnection network (IN) connects them. A complete interconnection such as crossbar may be cost prohibitive, but a shared-bus interconnection may be inefficient and unreliable. We need to design INs with both cost and performance between these two extremes.

Parallel or distributed computers can generally be divided into two categories: multiprocessors and multicomputers. The main difference between the two lies in the level at which interactions between the processors occur.

A multiprocessor system must permit all processors to directly share main memory, as shown in Figure 1a. In a multicomputer system, however, each processor has its own local memory. Sharing between the processors occurs at a higher level, through a complete file or data set. As shown in Figure 1b, a processor cannot directly access another processor's local memory.

Sculpture Simulation © Paul Brown/Digital Pictures 1983

An IN is a complex connection of switches and links that permits data communication between processors and memories in a multiprocessor system or between the processors in a multicomputer system.

A multiprocessor or a multicomputer architecture is further characterized by the topology of the IN it uses. Current multiprocessor organizations are based on crossbar, multistage interconnection networks (MINs) and multiple-bus networks, as shown by the first four articles in this special issue. Multicomputer architectures include topologies such as star, ring, tree, and hypercube, as shown by the last article in this issue.

Classification of INs

Generally speaking, any IN has three operational characteristics. These are based on its timing, switching, and overall control.

The timing control of an IN can be either synchronous or asynchronous. Synchronous systems are characterized by a central global clock that broadcasts the clock signal to all devices on the IN so that they operate in a lockstep fashion. Asynchronous systems, on the other hand support independent operation of the devices without a global clock.

An IN transfers data using either circuit switching or packet switching. In circuit switching, once a device is granted a path in the IN it will occupy that path for the duration of the data transfer. In packet switching, the information is broken into small packets that individually compete for a path in the IN.

Based on the overall control of the network, an IN may be classified as centralized or decentralized. In centralized control, a global controller receives all requests and transmits the messages in the IN. In a decentralized system, requests are handled independently by different devices in the IN.

0018-9162/87/0600-0009$01.00©1987IEEE

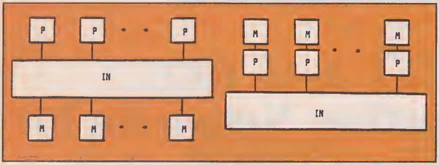


Figure 1. A multiprocessor architecture (a) and a multicomputer architecture (b).

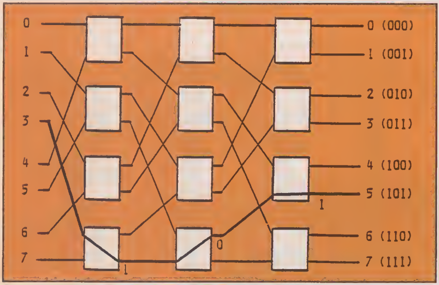


Figure 2. An 8 × 8 omega network.

These three operational characteristics with the topology define an IN.[1] For example, the Butterfly parallel processor uses an asynchronous, packet-switched, decentralized MIN.

Multiprocessor INs

Any processor in a multiprocessor system should be able to directly address every shared memory module through the IN. As a result, the performance of a multiprocessor rests primarily on the design of its IN.

A shared-bus interconnection is the least complex, but it does not allow more than one processor at a time to access a shared memory. A large number of processors means a long wait for the bus.

On the other hand, a crossbar supports all possible distinct connections between the processors and memories simultaneously. However, the cost of such a network is $O(N^2)$ for connecting N inputs to N outputs. For a system with hundreds of processors, the cost of such an IN is prohibitively high.

The cost and performance of MINs and multiple-bus networks hit a reasonable balance between those of a shared bus and a crossbar. Design, analysis, and development of MINs during the last decade have made them the most current technology.

The multiple-bus interconnection is somewhat new in the IN arena, as described in detail in the third article in this special issue.

An $N \times N$ MIN connects N processors to N memories. For N a power of two, it employs $\log_2 N$ stages of 2×2 switches with $N/2$ switches per stage. Each switch has two inputs and two outputs. The connection between an input and an output depends on a control bit c provided by the input. When $c = 0$, the input is connected to the upper output. When $c = 1$, it is connected to the lower output.

An omega network,[2] shown in Figure 2, is characterized by a perfect shuffle interconnection preceding every stage of switches. The requesting processor generates a tag that is the binary representation of the destination. The connection of a switch at the ith stage is then accomplished by the ith bit of this binary tag counted from the most significant bit. This self-routing property of an MIN avoids the need for a central controller, making it suitable for multiprocessors.

The connection between input 3 and output 5 (101_2) is shown by a bold line in Figure 2. Many significant MINs, such as SW Banyan, generalized cube, and base line,[1] have been proposed, perhaps confusing the community at large. However, all these networks are similar except for the interconnection between the adjacent stages.

The switch size in an MIN need not be restricted to 2×2. In fact, the Butterfly parallel processor connects N inputs to N outputs using 4×4 crossbar switches and $\log_4 N$ stages with $N/4$ switches per stage.

A delta network can connect $M = a^n$ inputs to $N = b^n$ outputs through n stages of $a \times b$ crossbar switches.[1]

The generalized shuffle network can connect any $M = m_1 * m_2 * \ldots * m_r$ to $N = n_1 * n_2 * \ldots * n_r$ outputs through r stages of switches.[3] The ith stage employs $m_i \times n_i$ crossbar switches and is preceded by a generalized shuffle interconnection that is essentially a superset of the omega and delta interconnections. This is the most generalized version of an MIN reported so far, in that it allows different input and output sizes.

Performance analysis shows that an MIN using 4×4 switches is more cost effective than one with 2×2 switches.[3]

Researchers recognized the advantages of MINs and started many academic and industrial research projects. University projects include TRAC at Texas, PASM at Purdue, Ultra-Computer at New York, and Cedar at Illinois. RP3 is a notable industry project at IBM and Butterfly is a successfully marketed product from BBN Laboratories Inc.

As these projects were starting, a serious drawback of MINs surfaced. There is only one path from an input to an output. It was necessary to incorporate some fault-tolerance into these networks so that at least a single fault in a switch or a link could be tolerated. This has given rise to an abundance of research during the past few years devoted to the design and evaluation of fault-tolerant MINs. The first two articles in this special issue are devoted to this relatively new area.

Multicomputer INs

Several structures for connecting a large multicomputer have been proposed in the literature.[1] These networks usually fall into the asynchronous, decentralized, packet-switched category. A dedicated link exists between two processors (nodes) over which the communication between them occurs. A message generated at a source node may go through a number of intermediate nodes in a store-and-forward manner before reaching the final destination.

Although several measures have been suggested in the literature, the characteristic of a symmetric IN can be easily expressed in terms of the degree of a node and the diameter of the IN. The degree of a node is the number of links per node. This reflects the cost associated with the network. The diameter is the maximum number of links a message has to travel between any source and any destination along the shortest path. Usually the networks that have a lower degree for each node give rise to a higher diameter, which means more delay in communication. A higher degree node reduces the diameter, but at a higher cost.

A bidirectional loop and completely connected structures represent these extreme examples, respectively.[1] The goal of research in the multicomputer IN area is, therefore, to design a network that has a low degree as well as a low diameter. The boolean n-cube or hypercube[4,5] represents a well-balanced structure between the above two extremes. It is described below.

A three-dimensional hypercube computer is shown in Figure 3. In general, a network with $N = 2^n$ nodes is arranged as an $n = \log_2 N$ dimensional hypercube with two nodes in each dimension. The nodes are represented by binary equivalents of the decimal numbers between 0 and $N-1$ where adjacent nodes differ by one bit. The distance between the source $X = x_{n-1}x_{n-2} \ldots x_1x_0$ and destination $Y = y_{n-1}y_{n-2} \ldots y_1y_0$ is equal to the number of bits by which X and Y differ in their binary representations. The diameter of this network is n and there exist n disjoint paths between a source and a destination. For example, in Figure 3 a message from 0 to 7 can be routed through any one of the following paths:

000 → 001 → 011 → 111
000 → 010 → 110 → 111
000 → 100 → 101 → 111

Hence the structure is highly fault-tolerant. The number of I/O ports per node is again $n = \log_2 N$, which many critics consider too high for large values of N. However, the structure is easily implementable, as evidenced by many hypercube products, such as the Intel iPSC, Ametek S/14, and NCUBE NCUBE/ten hypercube computers. Benchmarks on some of these computers are given in the last article of this special issue.

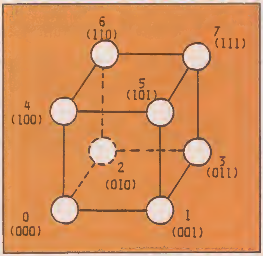


Figure 3. A hypercube network with eight nodes.

Bhuyan and Agrawal[5] extended the hypercube design to any number of nodes N, in contrast to $N = 2^n$ for traditional hypercubes. If $N = m_{n-1}*m_{n-2}* \ldots * m_1*m_0$ for integer values of m and n, an n-dimensional hypercube can be obtained with the ith dimension containing m_i nodes.

Analysis of such a generalized hypercube shows that it is more cost effective to build a system with four nodes in each dimension, compared to two in normal hypercubes.[4]

About this issue

The article by Adams, Agrawal, and Siegel in this issue is a first attempt to survey fault-tolerant networks. It compares the properties of different networks and provides the reader with the state of the art in this area. The second article, by Kumar and Reddy, focuses on a particular fault-tolerance technique that can be applied to shuffle exchange MINs. The reader, therefore, is exposed to the basic concept of fault-tolerance as applicable to MINs.

During the past three to four years, a lot of interest has developed in multiple-bus INs. Bus structures, in general, are easily understood. Multiple-bus systems can be viewed as an incremental expansion of many single-bus multiprocessor architectures. In the third article of this issue, Mudge, Hayes, and Winsor provide a tutorial on this new type of IN, including a hypothetical design of a multiple-bus multiprocessor.

The use of optics for communication purposes has been increasing dramatically, and its use in local area networks has received considerable attention. The article by Sawchuk, Jenkins, Raghavendra, and Varma provides different designs and discusses various trade-offs for the implementation of optical crossbar networks. Unlike their electronic counterparts, it seems feasible to build moderately large optical crossbars in a physically compact unit.

A number of structures or topologies such as loop, tree, full connection, and hypercube have been proposed to connect a network of computers. In the last article of this issue, Reed and Grunwald present a comparative study of these multicomputer networks. They also supplement their theoretical studies with benchmarks they obtained on a hypercube computer.

The development of this issue was guided by a desire to concentrate on interconnection network research that has emerged during the past five years. Thus we intend to bring the reader up-to-date since the last special issue on interconnection networks, which appeared in the December 1981 *Computer*, edited by C. L. Wu. I hope that readers will find the issue interesting. □

Acknowledgments

I would like to thank the authors and reviewers who helped to make this special issue a reality. They had severe time constraints and worked even during the Christmas holidays. Special thanks are due to Bruce D. Shriver, the editor-in-chief, for his constant guidance, encouragement, and help in the review process. Finally, my deep appreciation goes to the secretarial staff at The Center for Advanced Computer Studies for their timely help.

References

1. T.Y. Feng, "A Survey of Interconnection Networks," *Computer*, special issue on Interconnection Networks, Dec. 1981, pp. 12-27.
2. D.H. Lawrie, "Access and Alignment of Data in an Array Processor," *IEEE Trans. on Computers*, C-24, Dec. 1975, pp. 1145-1155.
3. L.N. Bhuyan and D.P. Agrawal, "Design and Performance of Generalized Interconnection Networks," *IEEE Trans. on Computers*, C-32, Dec. 1983, pp. 1081-1090.
4. H. Sullivan and T.R. Bashkow, "A Large Scale, Homogeneous, Fully Distributed Parallel Machine, I," *Proc. 4th Ann. Symp. on Computer Architecture*, IEEE 77CH 1182-5C, 1977, pp. 105-117.
5. L.N. Bhuyan and D.P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network," *IEEE Trans. on Computers*, C-33, April 1984, pp. 323-333.

Questions regarding this special issue should be addressed to Bhuyan at The Center for Advanced Computer Studies, University of Southwestern Louisiana, PO Box 44330, Lafayette, LA 70504.

Laxmi N. Bhuyan is an associate professor in The Center for Advanced Computer Studies at the University of Southwestern Louisiana, in Lafayette. His present research interests include parallel and distributed computer architecture, performance and reliability evaluation, and local area networks.

Bhuyan received his BS and MS in electrical engineering from the Regional Engineering College, Rourkela under Sambalpur University, India. He received the PhD in computer engineering from Wayne State University in Detroit in 1982.

Bhuyan is a senior member of the Computer Society.

---

acm

CALL FOR PARTICIPATION

THE COMPUTER SOCIETY
OF THE IEEE

Hypertext '87

SYSTEMS
APPLICATIONS
ISSUES

NOVEMBER 13-15, 1987

University of North Carolina
Chapel Hill, North Carolina USA

FOR ADDITIONAL INFORMATION CONTACT:

John B. Smith
Dept. of Comp. Sci.
UNC
Chapel Hill, NC 27514
919-962-5021
jbs@unc.csnet

Frank Halasz
MCC
9390 Research Blvd.
Austin, TX 78759
512-338-3648
halasz@mcc.csnet

DEADLINES:

August 1 POSITION PAPERS DUE
August 1 TECHNICAL PAPERS DUE
September 1 NOTICE OF WORKSHOP ADMITTANCE
September 15 NOTICE OF PAPER ACCEPTANCE
October 5 FINAL PAPERS FOR PREPRINTS DUE

KEYNOTE SPEAKER
Andries van Dam, Brown University

PROGRAM COMMITTEE:
Stephen F. Weiss, UNC, Program Co-chair
Mayer Schwartz, Tektronix, Program Co-chair
Greg Crane, Harvard University
Norman Delisle, Tektronix
Mark Frisse, Washington Univ. Medical School
Frank Halasz, MCC, Workshop Co-Chair
David Lowe, NYU
Norm Meyrowitz, Brown University
Theodor Nelson, Project Xanadu
Walter Scacchi, USC
John B. Smith, UNC, Workshop Co-Chair
Lucy Suchman, Xerox Parc
Randy Trigg, Xerox PARC
Andries van Dam, Brown University
Stephen A. Weyer, Apple
Nicole Yankelovich, Brown University, Pub. Chair

Co-Sponsored by ACM*, CS of the IEEE*, and UNC, with support from ONR, MCC and NSF*, in cooperation with ACM SIG CHI.
* Approval pending.

Now you can have perfect Fortran to C conversion ...at the touch of a button.

Here's your chance to take your entire Fortran library and convert it all to C.

Because FORTRIX™-C is here...the only software available that translates Fortran files to C code...automatically, perfectly and completely.

It doesn't matter which Fortran dialect, style or sequence you have...FORTRIX will convert it all. And because it's a machine translation, you get all the cost advantages of extraordinary speed. That's 600 lines a minute. And 50,000 lines that are ready to run in two weeks.

What's more, FORTRIX is proven. Right now it's on site and working worldwide at over 100 locations... including IBM,™ AT&T,™ Mitsubishi, U.S. Army and many others.

Find out how easy it is to keep your Fortran programs and still move up to C. Call our toll free number and we'll tell you exactly what FORTRIX will do for you. If it's more information you want, fill out this coupon and we'll send you technical details.

FORTRIX-C

The last word in Fortran to C conversion

Rapitech Systems Inc.
Montebello, Corporate Park
Suffern, NY 10901

(914) 368-3000 1-800-FORTRIX. Telex 509210

Name ______________________
Title ______________________
Company ______________________
Address ______________________
City ____________ State ______ Zip ______
Phone ______________________

FORTRIX—Registered Trademark of Rapitech Systems Inc. Other names indicated by TM are Trademarks of their respective companies.

Reader Service Number 1

A Survey and Comparison of

Fault-Tolerant Multistage Interconnection Networks

George B. Adams III, Research Institute for Advanced Computer Science
Dharma P. Agrawal, North Carolina State University
Howard Jay Siegel, Supercomputing Research Center*

Fault model and fault-tolerance criterion are essential metrics for any comparison of network fault-tolerance capabilities.

Today all branches of engineering and a multitude of other disciplines rely firmly and increasingly on computational support. However, each discipline has important problems that, to be solved feasibly, await our ability to field computers with orders of magnitude greater performance than currently available. Consequently, the need for very high performance computing is larger than ever and growing.

To date, high-performance computers have owed their speed primarily to advances in circuit and packaging technology. These technologies are subject to physical limits constraining the ultimate speed of a conventional uniprocessor computer. Parallel processing computers executing problem solutions expressed as parallel algorithms translated into parallel machine programs can exceed the single processor speed limit.

The means for communication among processors, memory modules, and other devices of a parallel computer is the interconnection network. Conventional von Neumann computers have relatively few subsystems to connect, yet a simple communication scheme such as a time-shared bus is an important system component. Large-scale multiprocessor computers have numerous communicating components, and therefore place great performance demands on an interconnection network.

Many interconnection networks for large-scale multiprocessor computer systems have been proposed.[1,2] Of these, *multistage interconnection networks* (MINs) compose a large subset, are well suited to communication among tightly coupled system components, and offer a good balance between cost and performance. For complex systems, assuring high reliability is a significant task. Thus, a crucial quality for MINs serving the communication needs of large-scale multiprocessor systems is fault-tolerance.

This article focuses on MINs that are intrinsically fault tolerant. Many have been proposed, and new ideas are published frequently. The survey herein, while not exhaustive due to space constraints, attempts to portray the diversity of fault-tolerant MINs and, through them, the scope of fault-tolerance techniques. The relative merits of fault-tolerant MINs can be difficult to assess. Therefore, once surveyed, the MINs are placed in perspective by defining a hypothetical MIN with ideal engineering characteristics and comparing the surveyed networks to this standard.

---

*This work was performed while Siegel was at the School of Electrical Engineering, Purdue University.

0018-9162/87/0600-0014$01.00©1987IEEE

Terminology

Figure 1 shows MIN hardware in the most general terms; dots indicate items that may repeat. MINs are comprised of a collection of *switches* and *links* between switches. A signal may enter or leave a network through a *port*. A network with A input ports and B output ports is an $A \times B$ network.

A switch may be viewed as a very simple network. Switches are multiport devices; the number of ports and the port-to-port connections supported within a switch vary among switch designs. A *crossbar* switch can simultaneously connect, in any pattern, a number of input/output port pairs equal to the minimum of the number of inputs and the number of outputs. A *selector* switch connects only one of its inputs to one of its outputs at a time.

The term *network component* may denote any element of the structure of a network. An interconnection network may consist of a single *stage*, or bank, of switches and may require that data pass through the network more than once to reach its destination.

A MIN is constructed from two or more stages of switches, and typically is designed so that data can be sent to the desired destination by one pass through the network.

A *fault-tolerant MIN* is one that provides service, in at least some cases, even when it contains a faulty component or components. A fault can be either permanent or transient; unless stated otherwise, we assume in this article that faults are permanent. Fault tolerance is defined only with respect to a chosen *fault-tolerance model*, which has two parts. The *fault model* characterizes all faults assumed to occur, stating the failure modes (if any) for each network component. The *fault-tolerance criterion* is the condition that must be met for the network to be said to have tolerated a given fault or faults.

Fault models may or may not correspond closely to predicted or actual experience with MIN hardware. In particular, a fault model may be chosen with characteristics that simplify reliability analysis, even if those characteristics depart widely from reality (such as assuming certain network components never fail). While fault-tolerance criteria typically closely reflect the normal (fault-free) operational capability of a network, this need not be so. The variability of fault-tolerance models hinders comparison of the engineering characteristics of fault-tolerant MINs.

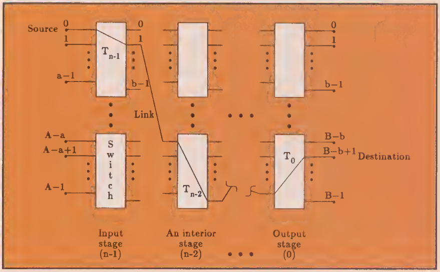


Figure 1. A generic MIN diagram detailing one path.

A network is *single-fault tolerant* if it can function as specified by its fault-tolerance criterion despite any single fault conforming to its fault model. More generally, if any set of i faults can be tolerated, then a network is *i-fault tolerant*. A network that can tolerate some instances of i faults is *robust* although not i-fault tolerant.

MINs can be considered from either a topological (graphical) or algebraic viewpoint. The *topology* of a network is the pattern of connections in its structure; the pattern can be represented by a graph. Topology is determined by switch design and the pattern of links. Different MINs are often compared graphically because comparison by topology is independent of hardware. When one network is said to be an instance of another, it is the network graphs being compared. Nodes in the graph of a MIN can be numbered, and then a MIN can be described in terms of the algebraic relations among the nodes. The algebraic model is useful in discussing control and communications routing strategy.

There are three basic forms of connection through a network. A *one-to-one connection* passes information from one network port, the *source*, to another network port, the *destination*. The exact route taken by the information is its *path*. Multiple one-to-one connections may be active simultaneously. A *permutation connection* is a set of one-to-one connections such that no two one-to-one connections have the same source or destination. Such connections are meaningful only in the context of networks with an equal number of sources and destinations. Information flow from one source simultaneously to two or more destinations is supported by a *broadcast connection*, and the route taken is a *broadcast path*.

Routing tags are a way of describing a path through a network and providing for distributed network control. For MINs, tags often take the form of a multidigit integer, each successive digit encoding the setting for the switch in the next stage along a desired path. Control is distributed if devices using the network generate their own routing tags and network switches can set themselves based on tag information. Figure 1 shows a switch in stages $n-1$, $n-2$, and 0 being set, respectively, by tag digits T_{n-1}, T_{n-2}, and T_0. Routing tags are particularly important for fault-tolerant MINs since they should be able to specify a functioning path if one exists; tag limitations translate into fault-tolerance limitations.

There are three methods for sources to generate routing tags that specify a fault-free path. With *nonadaptive routing* a source learns of a fault only when the path it is attempting to establish reaches the faulty network component. Notice of the fault is sent to the source, which tries the next alternative path. This approach requires little hardware, but may have poor performance. There are two forms of *adaptive routing*. With *notification on demand* a source maintains a table of faults it has encountered in attempting to establish paths and uses this

Sculpture Simulation © Paul Brown/Digital Pictures 1983.

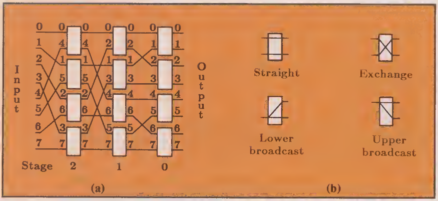


Figure 2. The generalized cube network for $N = 8$ (a) and four states of the generalized cube switching element (b).

information to guide future routing. With *broadcast notification* of a fault, all sources are notified of faulty components as they are diagnosed.

A fault-free path need not be specified by a source if routing tags can be modified in response to faults encountered as a path is followed or established. This *dynamic routing* can be accomplished in MINs constructed of switches capable of performing the necessary routing tag revisions.

Fault-tolerance issues

Fault tolerance can be achieved at various levels in a complex system. For example, when a process running on the TRAC[3] multiprocessor system encounters a fault in the portion of the TRAC MIN that links its set of system resources, it is dynamically rescheduled to another set of resources not affected by the MIN fault. The MIN subsystem is not itself intrinsically fault tolerant, but TRAC as a whole is, by allowing graceful degradation. Because fault-tolerant MINs are the focus of this article, we do not address fault tolerance of systems having embedded MINs.

Many fault-tolerant systems require fault diagnosis (detection and location) to achieve their fault tolerance. Techniques such as test patterns, dynamic parity checking, and write/read-back/verify for use in various MINs have been described. We assume fault diagnosis to be available as needed with respect to the surveyed MINs and do not discuss it further.

Techniques for fault-tolerant design can be categorized by whether they involve modifying the topology (graph) of the system. Three well-known methods that do not modify topology are error-correcting codes, bit-slice implementation with spare bit slices, and duplicating an entire network (this changes the topology of the larger system using the network).

These approaches to fault tolerance can be applied to MINs. A number of techniques have also been developed tailored closely to the nature of MINs and their use. The following survey explores these methods in particular.

Survey

The surveyed networks are ordered roughly by the hardware modifications made to provide redundancy, from less to more extensive. Many possible techniques exist, including adding an extra stage of switches, varying switch size, adding extra links, and adding extra ports. (The technique of *chaining* switches within a stage so that data can sidestep a faulty switch is discussed in detail elsewhere in this issue.[4])

Extra stage cube and delta networks. The extra stage cube (ESC)[5,6] network derives from the generalized cube MIN.[7] The generalized cube is an $N \times N$ MIN, $N = 2^n$, with $n = \log_2 N$ stages, each stage consisting of N links connected to $N/2$ switches (see Figure 2a). The links connecting stage i switches to stage i-1 switches are stage i links. Each switch, called an *interchange box*, is 2×2, individually controlled, and can be set to one of four states (see Figure 2b). The upper I/O ports on any switch have the same numeric labels; lower I/O ports also have the same labels. Upper and lower switch port labels in stage i differ only in the ith bit position.

The ESC is formed from the generalized cube by adding an extra stage to the input side of the network along with multiplexers and demultiplexers at the input and output stages, respectively. In addition, dual I/O links to and from the devices using the network are required. Stage n is connected like stage 0, that is, links that differ in the low order bit are paired. Figure 3 illustrates the ESC for $N = 8$.

Stage n and stage 0 can each be enabled or disabled (bypassed). A stage is *enabled* when its switches are being used to provide interconnection; it is *disabled* when its switches are being bypassed. Enabling and disabling in stages n and 0 are accomplished as shown in Figure 4. Having multiple ports per device using the ESC is equivalent to the input demultiplexer and output multiplexer arrangement originally described for the network.[5]

Normally, the network will be set so that stage n is disabled and stage 0 is enabled. The resulting structure matches that of the generalized cube. If a fault is found after running fault detection and location tests, the network is reconfigured. A fault in a stage n switch requires no change in network configuration; stage n remains disabled. If the fault occurs in stage 0, then stage n is enabled and stage 0 is disabled, i.e., stage n replaces the function of stage 0. For a fault in a link or in a switch in stages $n - 1$ to 1, both stages n and 0 will be enabled.

Enabling both stages n and 0 provides two distinct paths between any source and destination, at least one of which must be fault-free given any single fault in the network. ESC multiple-fault tolerance is enhanced by individually enabling and disabling stage n and 0 switches.[6]

The ESC fault model assumes that (1) any network component can fail, (2) faulty components are unusable, and (3) faults occur independently. The ESC fault-tolerance criterion is retention of the fault-free interconnection capability of a generalized cube MIN for one-to-one and broadcast connections. This corresponds to *full access*, the ability to connect any MIN input to any output. The ESC is single-fault tolerant and robust in the presence of multiple faults.

The ESC can be controlled by a simple extension of the routing tags used for the generalized cube. The tags are readily computed as the bit-wise Exclusive-OR of the number of the network input and output ports to be connected, perhaps modified by information as to the location of a fault (if any).

The augmented delta network[8] also utilizes an extra stage, without circuitry to enable and disable the input and output stages, to achieve fault tolerance. A delta, or *digit controlled*, network is an $a^w \times b^w$ network with w stages, each consisting of $a \times b$ crossbar switches, and with a link pattern between stages that provides a unique path of fixed length between any network input and output. Further, the link pattern is such that information can be routed from an input to an output using the switch output corresponding to a base-b digit in the base-b representation of the output number. The extra stage of N/b $b \times b$ switches results in b paths between any pair of network input/output ports. With a fault model assuming (1) any link and any interior switch can fail, (2) faulty components are unusable, and (3) faults occur independently, and a fault-tolerance criterion of retention of full access capability, an augmented delta network is $(b-1)$-fault tolerant.

Multipath omega network. The multipath omega network[9] derives from the omega MIN.[10] A $B^m \times B^m$ omega network consists of m stages of $B \times B$ crossbar switches linked by $(B*B^{m-1})$-shuffle interconnections, a generalized shuffle. If $B = 2$ and $m = \log_2 N$, the omega network is obtained; its topology is identical to that of the generalized cube. For the multipath omega network the fault model is the same as for the augmented delta; the fault-tolerance criterion is again full access.

Figure 5 shows one possible multipath omega network for $N = 16$. Its structure is described by the pseudofactorization $<4,2,2,4>$ of N. A *pseudofactorization* of N is an f-tuple $<B_1,B_2,\ldots,B_f>$ of integers with $B_1*B_2*\ldots*B_f = B$, such that B is a multiple of N. Let $B/N = R$, then an R-path multipath omega network corresponding to the pseudofactorization has f stages with stage i consisting of $B_i \times B_i$ crossbar switches. Links entering stage i switches implement the $\{k_i\, B_i * N/(k_i\, B_i)\}$-shuffle interconnection, where $k_i \geq 1$ and k_i is an integer chosen so that there are exactly R ways to connect any network input to any network output. For the network of Figure 5a, $k_1 = 1$, $k_2 = 1$, $k_3 = 8$, and $k_4 = 1$. The R paths in a multipath omega need not be distinct, hence, a single fault may affect more that one path. A multipath omega

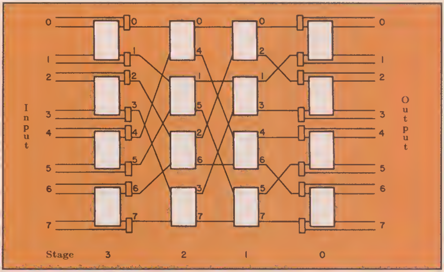


Figure 3. The extra stage cube network for $N = 8$.

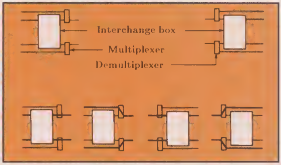


Figure 4. Detail of input stage switch with multiplexer for enabling and disabling (a), output stage switch with demultiplexer for enabling and disabling (b), input stage switch enabled (c), input stage switch disabled (d), output stage switch enabled (e), output stage switch disabled (f).

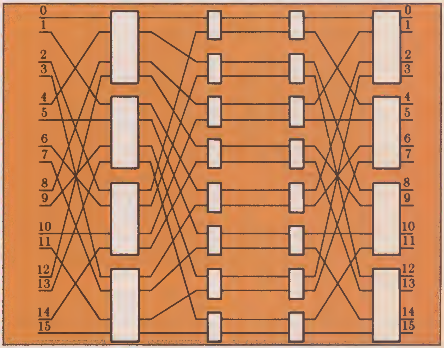


Figure 5. One possible multipath omega network for $N = 16$.

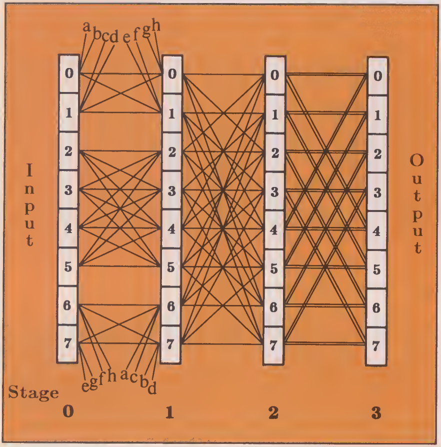
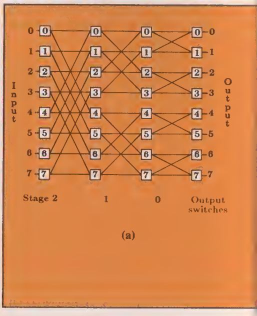


Figure 6. The F-network for $N = 8$.



(a)

network with λ distinct paths between a source/destination pair is $(\lambda - 1)$-fault tolerant.

Routing in the multipath omega network is controlled by routing tags and resembles the procedure used for the omega network and generalized shuffle network. The destination address determines n bits of the tag; r additional bits select a particular path out of the R alternatives. Stage i switch settings are controlled by $b_i = \log_2 B_i$ bits.

F-network. The F-network[11] is $2^n \times 2^n$ with $n + 1$ stages of $N = 2^n$ switches each, that are, in general, 4×4 selectors. A switch in stage j, W_j, is denoted by a bit string $W_j = w_{n-1} \ldots w_1 w_0$. It connects to the stage $j + 1$ switches $W_{j+1} = w_{n-1} \ldots w_1 w_0$, $X_{j+1} = w_{n-1} \ldots w_{j+1} \overline{w_j} w_{j-1} \ldots w_1 w_0$, $Y_{j+1} = \overline{w_{n-1} \ldots w_{j+1} w_j} w_{j-1} \ldots w_1 w_0$, and $Z_{j+1} = \overline{w_{n-1} \ldots w_{j+1}} w_j w_{j-1} \ldots w_1 w_0$. Figure 6 shows the F-network for $N = 8$. The F-network contains the structure of the generalized cube and can emulate it by using only the W_{j+1} and X_{j+1} connections. Thus, the fault tolerance approach of the F-network adds links (Y_{j+1} and Z_{j+1}) to the generalized cube structure. At each stage except the output stage, two different switches can be selected while maintaining the same destination. A faulty, as well as a busy, switch can be avoided by taking the appropriate path.

The F-network uses a yet less strict fault model than the augmented delta, and assumes (1) only interior switches fail, (2) failed switches are unusable, and (3) faults occur independently. The fault-tolerance criterion is retention of full access. The network is single-fault tolerant and robust in the presence of multiple faults.

Dynamic redundancy network. The dynamic redundancy network (DR)[12] is based on the graph representation of the generalized cube network. Figure 7a shows the graph representation of the generalized cube network of Figure 2a. Figure 7b relates the graph and interchange box representations.

The DR network has $N + S$ I/O ports and $\log_2 N$ stages, each with $N + S$ switches followed by $3(N + S)$ links. There are $N + S$ output switches. Each switch j at stage i of the network has three links to stage $i - 1$. One is connected to switch $(j - 2^i) \bmod (N + S)$, the second to switch j, and the third to switch $(j + 2^i) \bmod (N + S)$. The DR output switches are its output ports. Figure 7c shows a DR network for $N = 8$ and $S = 2$. The S spare I/O ports allow S spares of the devices using the network, thus providing fault tolerance for devices.

A *row* of a DR network contains all the switches having the same address, all links incident out of them, and the associated network input and output links. A row has the same address as its switches. Two rows of the DR network are said to be *adjacent* if their addresses are consecutive (*mod* $N + S$).

The DR fault model is the same as for the ESC: any component may fail. Its fault tolerance criterion is retention of the ability to perform any set of simultaneous connections possible with the generalized cube. This is a stricter criterion than used for the ESC.

When there are no faults, rows 0 to $N - 1$ are used to emulate the generalized cube network. (For example, Figure 7a is a subgraph of rows 0 to 7 of Figure 7c.) If a component of row j is found to be faulty, the network is reconfigured so that the

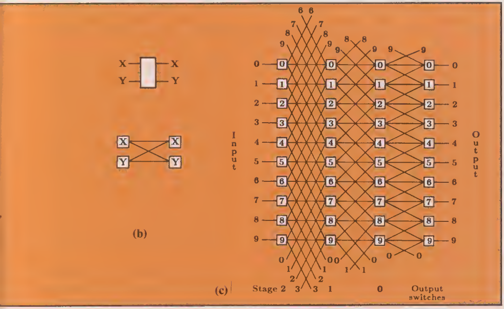


Figure 7. Graphical representation of the generalized cube of Figure 2a (a), relationship between graphical representation and interchange box representation (b), and DR-network for $N = 8$ and $S = 2$ (c).

switches physically numbered p are logically renumbered $t(p)$, where $t(p) = (p - j - S) \bmod (N + S)$. As long as N adjacent rows remain, the DR network can contain the generalized cube subgraph and thus act as a fault-tolerant generalized cube.

Enhanced IADM network. As depicted in Figure 8, the IADM[13] consists of $n = \log_2 N$ stages each with N switches and $3N$ links, plus one column of output switches that are the network output ports. Each switch is a 3×3 selector. At stage i, $0 \le i < n$, switch j, $0 \le j < N$, is connected to switches $(j - 2^i) \bmod N$, j, and $(j + 2^i) \bmod N$ in stage $i + 1$. These links are known as the minus, straight, and plus links, respectively. Since $j - 2^{n-1} \equiv (j + 2^{n-1}) \bmod N$, there actually exist just two distinct logical data paths from each stage $n - 1$ switch.

Unlike the generalized cube, delta, and omega networks, the IADM can tolerate some faults (given a fault-tolerance criterion of full access) because of multiple paths between a source, S, and destination, D, if $S \neq D$. The nature of this native fault tolerance has been studied in detail for the closely related gamma network. The extra-stage gamma[14] is single-fault tolerant with respect to the fault model and fault-tolerance criterion of the augmented delta.

The enhanced IADM has the same fault model as the augmented delta network, as well as the same fault-tolerance criterion.

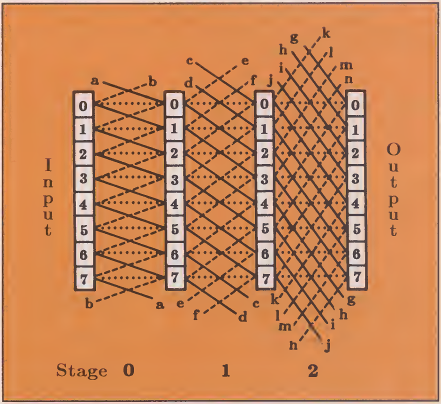


Figure 8. The inverse augmented data manipulator network for $N = 8$.

There are two forms of enhanced IADM. One version has redundant straight links. Faulty links can be avoided by using the second straight link, but switch faults cannot.[13] A second, more effective enhanced IADM adds *half links* to each of stages 1 through $n - 1$. Half links connect a switch m in stage i to switches

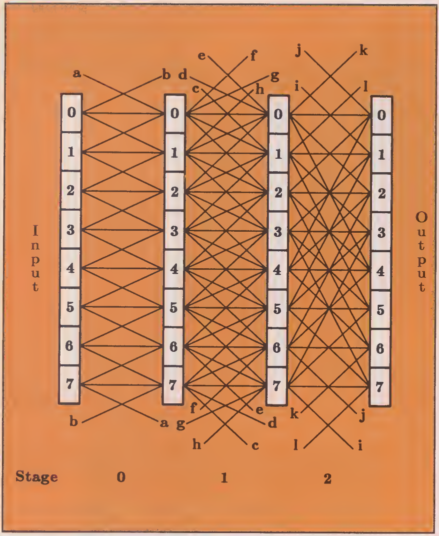


Figure 9. The enhanced inverse augmented data manipulator network with half links for $N = 8$.

$(m + 2^{i-1})$ *mod N* and $(m - 2^{i-1})$ *mod N* (see Figure 9).

Adding half links provides single-fault tolerance to any switch or link failure because at any switch (except those in stage $n - 1$, the last stage) there exist at least two (sometimes three) links leading to distinct switches in the successive stage, any of which can be used to satisfy the routing need. With a single-stage look-ahead technique and dynamic routing, the network is two-fault tolerant.[13] Each switch must have the ability to determine if the switches or links to which it connects are faulty and, if so, to modify routing tags to avoid a faulty component.

β-networks. A *β-network* results from simply interconnecting a set of β-elements; no particular topology is specified.[15] A *β-element* is an interchange box that can perform only straight and exchange permutations.

The β-network fault model assumes (1) only switches fail, by becoming stuck in one of their two allowed states, (2) faulty switches are usable, and (3) faults occur independently. A β-network is said to tolerate a fault if the fault does not destroy dynamic full-access capability. A β-network is defined as having the *dynamic full-access* property if any given network input can be connected to any single network output in a finite number of passes through the network. Between passes it is assumed that each output can connect to its corresponding input (i.e., the input with the same number as the output) via a path outside the network. This is a considerably less restrictive fault-tolerance criterion than those of the other networks surveyed.

Augmented C-network. The augmented C-network (ACN)[16] fault-tolerance technique involves doubling the number of switch input and output ports, maintaining the same type switch function, and then adding links to connect selected pairs of switches in adjacent stages. The ACN derives from the C-network, an $N \times N$ MIN having an arbitrary number of stages of $N/2$ 2×2 crossbar switches each. Stages are numbered 0 to $m - 1$ from input to output.

Let one output of a switch be labeled 0 and the other, 1. For a switch P in stage i, i not the output stage, its *0-successor*, denoted $succ^0(P)$, is the switch in stage $i + 1$ connected to its 0 output. The *1-successor*, $succ^1(P)$, is the stage $i + 1$ switch connected to its 1 output.

The topology of a C-network is defined by the following relationship. For each switch P_j in stage i, $0 \leq i < m - 1, 0 \leq j < N/2$, there exists a switch P_k, $0 \leq k < N/2$ and $k \neq j$, in stage i such that $succ^0(P_j) = succ^0(P_k)$ and $succ^1(P_j) = succ^1(P_k)$. Switches P_j and P_k are said to be *conjugate*, denoted as $conj(P_j) = P_k$ and $conj(P_k) = P_j$. For example, in the generalized cube of Figure 2a the switch with labels 0 and 4 and the switch with labels 2 and 6 are conjugate.

The notion of conjugate switches has been discussed using the term *output buddies*.[17] The concept of *input buddies*, the 0- and 1-successors of both a switch and its conjugate, was also noted. A network in which two pairs of input buddies also constitute two pairs of output buddies, such as the generalized cube, has the *strict buddy* property. C-networks do not necessarily have this property.

A C-network with the additional property that $succ^0(P) \neq conj(succ^1(P))$ and $succ^1(P) \neq conj(succ^0(P))$, such as the generalized cube, provides the basis for the ACN. So that we can compare all networks in this survey on an equal basis, assume the C-network to have $\log_2 N$ stages. Beginning with such a C-network, an ACN is constructed by replacing all the 2×2 switches with 4×4 crossbar switches with inputs labeled 0, 1, *conj*(0), and *conj*(1). Switch outputs are labeled similarly. Those switch inputs and outputs labeled 0 and 1 are connected exactly as in the base C-network. The *conj*(0) and *conj*(1) ports are connected as shown in Figure 10.

One stringent ACN fault model is that

(1) any network component can fail, and (2) faulty components are unusable. Except for not specifying fault independence, this is the same stringent fault model used by the ESC. As does the ESC, the ACN requires dual I/O ports for each device using the network. The ACN fault-tolerance criterion is retention of full access capability. The ACN provides 2^n distinct paths between any source and destination (at each stage either switch *i* or switch *conj*(*i*) can be used), but most of these paths are not disjoint. In general, there exist at least two switches at any stage and two links between stages by which a given source and destination can be joined. Thus, the ACN is single-fault tolerant to both switch and link failures.

Routing in the ACN is predicated on a routing tag scheme existing for the particular base C-network. We assume that a switch can determine when a successor switch is faulty. When no faults or busy switches occur, the tag is determined and interpreted by the ACN as it would be for the base C-network. Otherwise, two proposed routing strategies utilize both the standard path and conjugate path switches.[16]

Merged delta network. An $N \times N$ merged delta network (MDN)[16] results from cross-linking corresponding stages of C copies of an $N/C \times N/C$ delta network. An MDN denoted by C-MDN indicates explicitly the number of copies. The basic switch for the MDN is a $2C \times 2C$ crossbar, with $\log_2 N/C$ stages. The construction procedure used for MDNs works with any MIN; the delta was chosen by the MDN developers for its simple routing scheme.

Number the C copies of the delta network 0 to $C - 1$ and denote source S, destination D, and switch P associated with copy c, $0 \leq c \leq C - 1$, as (S, c), (D, c), and (P, c), respectively. Let $0 \leq j \leq N/2$. If source S connects to switch $P_{j,0}$ in the input stage of the delta network, then sources (S, c) connect to input stage switches $(P_{j,0}, c)$ in the MDN. If destination D connects to output stage switch $P_{j,n-1}$ in the delta network, then destinations (D, c) connect to switches $(P_{j,n-1}, c)$ in the MDN. Finally, if the outputs of stage i switch $P_{j,i}$ connect to stage $i + 1$ switches $P_{k,i+1}$ and $P_{m,i+1}$ in the delta network, then in the MDN each switch $P_{j,i}$, c connects to switches $P_{k,i+1}$, c and $P_{m,i+1}$, c. Figure 11 shows a 2-MDN ($C = 2$) for $N = 8$. Bold lines denote links from the replicated networks; light lines denote

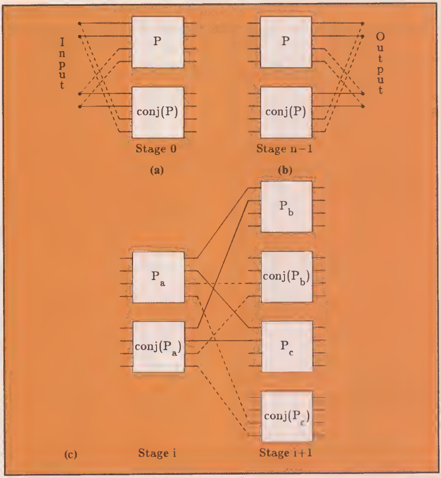


Figure 10. Connections to stage 0 switches in an ACN (a), connections from stage $n - 1$ in an ACN (b), connections from a conjugate switch pair in an ACN (c).

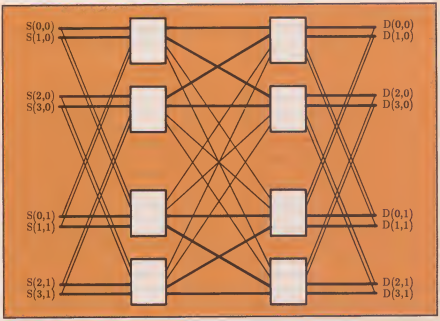


Figure 11. MDN for $N = 8$ and $C = 2$.

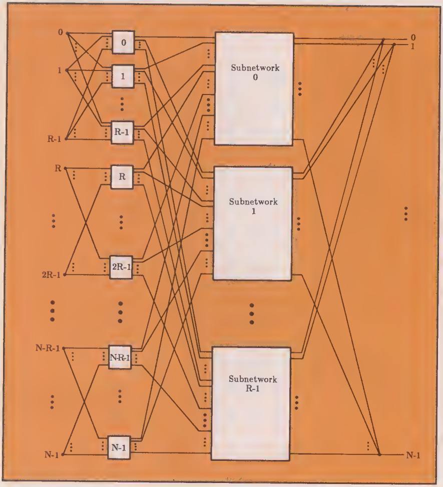


Figure 12. INDRA network shown as the union of R subnetworks.

links merging the two copies.

The MDN fault model assumes (1) only switches fail, and (2) faulty switches are unusable. The fault-tolerance criterion is full access. The MDN provides $C^{\log_2(N/C)}$ paths between any source and any destination (at each stage the corresponding switch in any of the C copies can be used). In general, there are at least C switches at any stage and C links between stages by which a given source and destination can join. Thus, the MDN is $(C - 1)$-fault tolerant to both switch and link failures (although the fault model assumes links never fail). Like the ESC and ACN, the MDN requires C I/O ports per device using the network.

MDN routing is based on the delta network routing procedure. Information is initially passed to a non-faulty switch in stage 0. From there, each switch in stage j forwards the information by choosing with equal probability one of the copies. If the selected switch is faulty another copy is chosen. At stage $n - 1$ the switch output is chosen to reach the intended destination.

INDRA network. Interconnection networks designed for reliable architectures, or INDRA[18] networks, incorporate multiple copies of a basic network, typically the omega network, but preface the copies with a distribution stage of switches (see Figure 12). INDRA networks have $N = 2^m$ inputs and outputs with $\log_R N + 1$ stages of $R \times R$ switches. Stages are numbered 0 through n from input to output, where $n = \log_R N$, and each stage has N switches (typically N is a power of R, but it need not be). They are connected by links in an $R*(N/R)$-shuffle. Other link interconnection patterns, including the cube interconnections, are possible. As with the MDN, the construction technique used for INDRA networks can be applied to any MIN.

INDRA networks can be conceived as the union of R parallel networks each with $\log_R N$ stages of $R \times R$ switches, with an initial distribution stage at the input. Like the ESC, ACN, and MDN, the INDRA network requires C I/O ports per device using the network.

The INDRA network fault model matches that used for the MDN with the addition that faults are explicitly assumed to occur independently. The fault-tolerance criterion is the ability to perform any permutation in one pass that an omega network of size N with $R \times R$ switches can perform in one pass.

The INDRA network is digit-controlled in the same sense as delta networks and performs all omega network-performable permutations, as well as others. Without using the R redundant links to stage 0, there exist R paths between any source/destination pair. Using redundant links yields R^2 paths, although they are not all disjoint. In the case of a stage 0 switch fault, data can be routed through one of the alternate links from the input port. Other switch faults can be avoided by taking one of the R^2 paths that exist between any source and destination that does not include the faulty switch. This is always possible, as any faulty switch not in stage 0 lies within some subnetwork (see Figure 12), and a path exists between any source and destination that passes through some other subnetwork. Faults are avoided either by having sources keep track of failed switches and avoid them when generating routing tags, or by a form of *simulcasting*, whereby information is sent on multiple paths simultaneously to one destination.[18]

The INDRA network is $(R - 1)$-fault tolerant. Like the MDN, the INDRA network can tolerate link faults.

Comparison

Table 1 summarizes the network fault tolerance information presented in our survey. It lists: the possible faults that can occur in each network under the assumed fault model; whether or not faulty components are usable; the fault-tolerance criterion; the method by which the network copes with faults; whether the network is single-fault tolerant; and how the network performs with multiple faults. The phrase "interior switch only" used in the table is another way of saying input and output switches are assumed fault-free.

Table 1. Summary of fault tolerance information for the surveyed networks.

Network	Fault Model	Fault-Tolerance Criterion	Fault-Tolerance Method	Single-Fault Tolerant*	Multiple-Fault Tolerant*
Extra Stage Cube	Any component; unusable	Full access	Alternate route	Yes	Robust
Augmented Delta	Interior switch or link; unusable	Full access	Alternate route(s)	Yes	$(b-1)$-fault tolerant with $b \times b$ switches
Multipath Omega	Interior switch or link; unusable	Full access	Alternate route(s)	Yes	$(\lambda-1)$-fault tolerant
F-network	Interior switch only; unusable	Full access	Alternate routes	Yes	Robust
Extra Stage Gamma	Interior switch or link; unusable	Full access	Possible alternate route	No; only robust	Robust
Dynamic Redundancy	Any component; unusable	Full access	Alternate (spare)	Yes	Limited robustness
Enhanced IADM (half links)	Interior switch or link; unusable	Full access	Alternate routes	Yes	Two-fault tolerant with lookahead
β-networks	Switch stuck, but usable	Dynamic full access	Repeated passes	Depends on topology	Typically robust
Augmented C-network	Any component; unusable	Full access	Alternate routes	Yes	Robust
Merged Delta	Any switch; unusable	Full access	Alternate routes	Yes	$(C-1)$-fault tolerant
INDRA	Any switch; unusable	Ω permutation in one pass	Alternate routes	Yes	$(R-1)$-fault tolerant

*Using the fault model and fault-tolerance criterion defined for the network in question.

Table 2 summarizes structural characteristics of the networks. Switch type, number of stages of switches, number of switches per stage, and number of links between stages of switches are four parameters relevant to both implementation cost and achievable fault tolerance. Networks with high switch and link counts may require high package counts and hence cost more to produce than less complex structures. Networks with high package counts are, all other things equal, more likely to experience failure during a given period of time. The nature of real failures must be taken into account, along with expected reliability, package count, and network topology and operation protocol to estimate a cost/fault-tolerance performance ratio for a particular network.

Despite the growing research on fault-tolerant MINs, the results often have limitations, including unreasonably optimistic fault-tolerance models, and increased data routing complexity. To place the surveyed networks in perspective, we compare them to a hypothetical fault-tolerant MIN having the following ideal engineering characteristics:

(1) Fault model—any network component can fail, and failed components are unusable.
(2) Fault-tolerance criterion—full access.
(3) Routing complexity—as low as any surveyed network.
(4) Hardware complexity—as low as any surveyed network.
(5) Fault-tolerance capability—single-fault tolerant and robust with respect to multiple faults.

Table 3 summarizes the comparison in qualitative terms. For example, in the column on fault model the phrase "slightly less strict" generally means input and output stage switches are assumed fault-free, while the phrase "less strict" typically

Table 2. Summary of structural characteristics of the surveyed networks.

Network	Network Ports per Source or Destination	Switch Type*	Number of Stages	Number of Switches/Stage	Number of Links/Stage
Extra Stage Cube	2	Four-state interchange box	$1+\log_2 N$	$N/2$	N
Augmented Delta	1	$a \times b$ crossbar	$i+\log_b N$, typically $i=1$	N/b	N
Multipath Omega	1	$B_i \times B_i$ crossbar in stage i	f, selectable within constraints	n/B_i in stage i	N
F-network	1	4×4 mux/demux	$1+\log_2 N$	N	$4N$
Extra Stage Gamma	1	3×3 crossbar	$2+\log_2 N$	N	$3N$
Dynamic Redundancy	1	3×3 mux/demux	$\log_2 N$	$N+S$	$3(N+S)$
Enhanced IADM (half links)	1	5×5 mux/demux	$1+\log_2 N$	N	$5N$ interior, $3N$ input stage, $4N$ output stage
β-networks	1	Two-state interchange box	Variable	Typically $N/2$†	Typically N†
Augmented C-network	2	4×4 crossbar	Same as underlying C-network	$N/2$	$2N$
Merged Delta	C	$2C\times 2C$ crossbar	$\log_2 (N/C)$	$N/(2C)$	CN
INDRA	R	$R\times R$ crossbar	$1+\log_R N$	N	RN

*Input and output stage switches may differ; see text.
†Depends on specific instance of network topology.

includes the previous restriction and adds the assumption of fault-free links.

The choice of fault model and fault-tolerance criterion plays a key role in determining the fault-tolerance characteristics of a network. As seen from Table 3, the comparison fault model is stricter, and likely more realistic, than that for all but three of the networks: the ESC, DR, and ACN. The comparison fault-tolerance criterion is the same as that for most of the networks surveyed. Table 3 indicates the more strict fault-tolerance criteria of the DR and INDRA networks; each requires some permutation connection capability in addition to full access.

The fault-tolerance criterion used to study β-networks offers a much less strict test to pass. It requires only the ability to connect any input to any output in a finite number of passes through the network. Successive passes are performed by returning data from a network output to the same numbered input. Even a fault-free β-network may require multiple passes for data to reach its destination, so the chosen fault-tolerance criterion is appropriate. However, the breadth of the class of β-networks makes it important to note that this forgiving criterion may inflate the capabilities attributed to more complex β-networks.

For most of the networks, routing in the presence of faults is little more complex than in the absence of faults. The notable exception to this is β-networks. Once faults have been detected and located, considerable work remains to determine if dynamic full access has been retained. Agrawal and Leu have developed a specific testing technique.[19] Of course, we can precompute the network status for an assumed fault condition. Given an input and output to be connected, binary routing tags will suffice. However, the dynamic full access procedure requires choosing a set of intermediate outputs reachable consecutively. We do not know of a general solution to this problem. In a practical sense, this may result in delivered fault tolerance less than that theoretically available.

The hardware complexity of the networks varies significantly, as shown in Table 2. The hardware complexity entries

Table 3. Comparison of surveyed networks with hypothetical network of ideal engineering characteristics described in text. Entries give the relationship between the network in question and the hypothetical network as regards a particular attribute.

Network	Fault Model	Fault-Tolerance Criterion	Routing Complexity	Hardware Complexity	Fault-Tolerance Capability*
Extra Stage Cube	As strict	Same	Comparable	Slightly greater	Comparable
Augmented Delta	Slightly less strict	Same	Comparable	Same	Comparable
Multipath Omega	Slightly less strict	Same	Comparable to slightly greater	Same	Comparable to greater†
F-network	Less strict	Same	Comparable	Slightly greater	Comparable
Extra Stage Gamma	Slightly less strict	Same	Comparable	Greater	Comparable
Dynamic Redundancy	As strict	More strict	Comparable	Slightly greater	Less
Enhanced IADM (half links)	Slightly less strict	Same	Comparable; complexity hidden in switch	Greater	Greater if complex routing used
β-networks	Much less strict	Much less strict	Much higher	Less to greater†	Comparable to greater†
Augmented C-network	As strict	Same	Comparable; complexity hidden in switch	Much greater	Comparable
Merged Delta	Less strict	Same	Comparable; complexity hidden in switch	Much greater	Greater
INDRA	Less strict	More strict	Comparable	Much greater	Greater

*Using the fault model and fault-tolerance criterion defined for the network in question.
†Depends on specific instance of network topology.

of Table 3 are qualitative. The intent is to reflect a rough estimate of package count at various levels, specifically, the chip and board level or multiple-chip carrier level. However, hardware complexity and implementation cost may bear little relationship to each other; knowledge of implementation details arising from a hardware design study is necessary for high-confidence estimates of cost. VLSI technology often allows a large reduction in package count, for example. With this caveat, the hardware complexity data suits our comparison purposes.

The fault-tolerance capabilities of the networks are all quite similar, given the individual standards by which we measure each network. This is apparent in the column on fault-tolerance capabilities in Table 3. However, because we study each network using its own fault model and fault-tolerance criterion, significant differences in capabilities might appear if we adopted a common fault model and fault-tolerance criterion.

The fault model and fault-tolerance criterion of the hypothetical network can be applied to the surveyed networks to compare their fault tolerance. A common fault model and fault-tolerance criterion form a single scale by which we can measure any network. The results of this common measure appear in Table 4, Column I. Only the ESC, DR, ACN, merged delta, and INDRA networks are single-fault tolerant with respect to this fault model and fault-tolerance criterion. The DR network requires connections so that the N adjacent rows remain accessible regardless of the location of a failing row.

Many of the networks fail to be single-fault tolerant because they cannot tolerate an input or output switch fault, as can the above five networks. Thus many fault models refer only to interior switch faults. If we amend the hypothetical network fault model to assume fault-free switches in the input and output stages, some of the networks are by this measure single-fault tolerant (see Table 4, Column II). Alternatively, the MINs that become single-fault tolerant under the relaxed fault model could be fitted with input and output stage bypass circuitry and become

Table 4. Fault tolerance capabilities of the surveyed networks assuming a common network fault model and fault tolerance criterion. I. Single-fault tolerant using hypothetical network fault model and fault-tolerance criterion (see text). II. Single-fault tolerant if hypothetical network fault model is relaxed to assume input and output stage switches are fault-free.

Network	I	II
Extra Stage Cube	Yes	Yes
Augmented Delta	No	Yes
Multipath Omega	No	Yes*
F-network	No	Yes
Extra Stage Gamma	No	Yes
Dynamic Redundancy	Yes	Yes
Enhanced IADM (half links)	No	Yes
β-networks	No	Yes*
Augmented C-network	Yes	Yes
Merged Delta	Yes	Yes
INDRA	Yes	Yes

*Typically yes, but depends on specific instance of network topology.

single-fault tolerant under the more stringent conditions.

The ESC was designed with a fault model and fault-tolerance criterion identical to that of the hypothetical network. Its routing complexity is low, based on easily computed tags that specify one of two possible paths between a source and destination. The ESC incorporates slightly more hardware than the simplest of any of the surveyed networks (such as the augmented delta with $b = 2$), having enable/disable circuitry for the input and output stage switches.

A β-network is single-fault tolerant given the relaxed hypothetical fault model if it provides at least two paths that share no network components other than their first and last β-elements between any source and any destination. This restriction must hold for all passes through the network that may be required for the two paths.

The ACN network is single-fault tolerant under the hypothetical fault model and fault-tolerance criterion. However, this capability comes at the price of greater hardware complexity. The ACN 4×4 crossbar switches are more complex than the switches of the hypothetical network and must have added circuitry in order to support the routing scheme. Further, the ACN has twice as many links as the hypothetical MIN.

The merged delta and INDRA networks are also single-fault tolerant under the assumptions made for Table 4. Again, this performance comes at the expense of hardware complexity. The MDN contains a total of $(N/(2C)) * \log_2(N/C)$ $2C \times 2C$ crossbar switches, each with significant routing support circuitry, and CN links. The INDRA network has $(1 + \log_R N) * N$ $R \times R$ crossbar switches and RN links. However, the MDN and INDRA networks provide C-fault and R-fault tolerance, respectively, and in the case of the INDRA network, with respect to a demanding fault-tolerance criterion. Given sufficiently high inherent component reliability, these networks may provide reliable interconnections in particularly demanding applications for which their probable high implementation costs can be justified.

We have discussed eleven fault-tolerant MINs: the extra stage cube, augmented delta, multipath omega, F-network, extra stage gamma, dynamic redundancy, enhanced inverse augmented data manipulator, β-networks, augmented C-network, merged delta, and INDRA. The MIN characteristics described included fault model, fault-tolerance criterion, fault-tolerance method, single- and multiple-fault tolerance, routing complexity, and hardware complexity. Fault model and fault-tolerance criterion are essential metrics for any comparison of network fault-tolerance capabilities. Routing and hardware complexity are key data for engineering trade-off decisions. These networks represent the wide range of topologies, switch design, and data routing procedures proposed for fault-tolerance.

Error correcting codes, connecting redundant subsystems in parallel, and dividing a subsystem into identical parts connected in parallel are all venerable techniques for achieving fault tolerance. The surveyed networks serve to illustrate fault-tolerance techniques developed or employed for MINs. Those techniques independent of network topology discussed here are

(1) simulcasting (multiple transmission of the same information on distinct paths) and
(2) dynamic full access (multiple passes through a MIN; used in β-networks).

Techniques that alter the topology of a MIN for better fault tolerance (and fault tolerant MINs embodying a given technique) are

(1) adding an extra stage with or without bypassing (ESC, augmented delta, extra stage gamma),
(2) adding extra links (F-network),
(3) increasing the number of network ports slightly (DR),
(4) completing inherent partial redundancy in a MIN (enhanced IADM),
(5) varying switch size and number of stages in an interrelated way (multipath omega),
(6) increasing switch size and adding corresponding links (ACN),
(7) replicating a MIN and cross-linking corresponding stages of switches (MDN), and
(8) replicating a MIN and adding an extra stage (INDRA).

We compared the surveyed networks by choosing a common fault model and fault-tolerance criterion based on the realistic representation of likely hardware failures rather than on easy analysis of reliability. The common fault model assumed that any network component could fail and, if faulty, was unusable. The fault-tolerance criterion specified that a network tolerated a fault only if it could still connect any one of its inputs to any one of its outputs.

Only five of the surveyed networks are single-fault tolerant under the stringent assumptions used for network comparison: the extra stage cube, dynamic redundancy, augmented C-network, merged delta, and INDRA. However, ACN, merged delta, and INDRA hardware complexity often results in a more complex and, hence, costly implementation than for the ESC. Data routing through these three networks is also more involved if their fault-tolerance capabilities are to be fully available. In their favor, all three are more tolerant of multiple faults. Finally, the dynamic redundancy network is most readily used in a system environment having spare processors to which processes can be rescheduled after failure blocks other processors from using the network.

Most of the other networks were single-fault tolerant if we relaxed the common fault model to assume perfectly reliable input and output stage switches. The ESC copes with input and output stage switch faults by having bypass circuitry that allows data to be routed past such faults when they occur. The ACN provides two links to two switches from each network input and two links to each output from two switches. The merged delta and INDRA networks each provide links to two or more switches from each network input and output. The other networks could achieve the same level of single-fault tolerance by either adding bypass circuitry in an appropriate way or including multiple connections between each network input and output port and network switches.

The field of fault-tolerant MINs is far from maturity. We can expect to see new fault-tolerant MINs and fault-tolerance techniques. Application of fundamental metrics such as fault model and fault-tolerance criterion will allow computer architects to continue to evaluate such networks when they are proposed.□

Acknowledgments

The authors thank G.J. Lipovski, M.R. Raugh, and D. Kaur for helpful discussions and comments.

This research was supported by Cooperative Agreement NCC 2-387 between the National Aeronautics and Space Administration and the Universities Space Research Association; the US Army Research Office, Dept. of the Army, under Contracts DAAG29-82-K-0101 and DAAG29-85-K-0236; the Rome Air Development Center, under Contract F30602-83-K-0119; and the National Science Foundation under Grant ECS 80-16580.

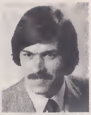
References

1. T.Y. Feng, "A Survey of Interconnection Networks," *Computer*, Dec. 1981, pp. 12-27.
2. H.J. Siegel, *Interconnection Networks for Large Scale Parallel Processing: Theory and Case Studies*, Lexington Books, Lexington, Mass., 1985.
3. M.C. Sejnowski et al., "An Overview of the Texas Reconfigurable Array Computer," *AFIPS 1980 NCC*, AFIPS Press, Montvale, N.J., 1980, pp. 631-641.
4. V.P. Kumar and S.M. Reddy, "A Fault-Tolerant Technique for Shuffle-Exchange Multistage Networks," *Computer*, this issue.
5. G.B. Adams III and H.J. Siegel, "The Extra Stage Cube: A Fault-Tolerant Interconnection Network for Supersystems," *IEEE Trans. Computers*, May 1982, pp. 443-454.
6. G.B. Adams III and H.J. Siegel, "Modifications to Improve the Fault Tolerance of the Extra Stage Cube Interconnection Network," *1984 Int'l Conf. Parallel Processing*, Computer Society Press, Silver Spring, Md., 1984, pp. 169-173.
7. H.J. Siegel and R.J. McMillen, "The Multistage Cube: A Versatile Interconnection Network," *Computer*, Dec. 1981, pp. 65-76.
8. D.M. Dias and J.R. Jump, "Augmented and Pruned *N*log*N* Multistage Networks: Topology and Performance," *1982 Int'l Conf. Parallel Processing*, Computer Society Press, Silver Spring, Md., 1982, pp. 10-11.
9. K. Padmanabhan and D.H. Lawrie, "A Class of Redundant Path Multistage Interconnection Networks," *IEEE Trans. Computers*, Dec. 1983, pp. 1099-1108.
10. D.H. Lawrie, "Access and Alignment of Data in an Array Processor," *IEEE Trans. Computers*, Dec. 1975, pp. 1145-1155.
11. L. Ciminiera and A. Serra, "A Connecting Network with Fault Tolerance Capabilities," *IEEE Trans. Computers*, June 1986, pp. 578-580.
12. M. Jeng and H.J. Siegel, "A Fault-Tolerant Multistage Interconnection Network for Multiprocessor Systems Using Dynamic Redundancy," *6th Int'l Conf. Distributed Computing Systems*, Computer Society Press, Silver Spring, Md., 1986, pp. 70-77.
13. R.J. McMillen and H.J. Siegel, "Performance and Fault Tolerance Improvements in the Inverse Augmented Data Manipulator Network," *9th Symp. Computer Architecture*, Apr. 1982, pp. 63-72.
14. K. Yoon and W. Hegazy, "The Extra Stage Gamma Network," *13th Symp. Computer Architecture*, Computer Society Press, Silver Spring, Md., 1986, pp. 175-182.
15. J.P. Shen and J.P. Hayes, "Fault-Tolerance of Dynamic-Full-Access Interconnection Networks," *IEEE Trans. Computers*, Mar. 1984, pp. 241-248.
16. S.M. Reddy and V.P. Kumar, "On Fault-Tolerant Multistage Interconnection Networks," *1984 Int'l Conf. Parallel Processing*, Computer Society Press, Silver Spring, Md., 1984, pp. 155-164.
17. D. P. Agrawal, "Graph Theoretical Analysis and Design of Multistage Interconnection Networks," *IEEE Trans. Computers*, July 1983, pp. 637-648.
18. C.S. Raghavendra and A. Varma, "INDRA: A Class of Interconnection Networks with Redundant Paths," *1984 Real-Time Systems Symp.*, Computer Society Press, Silver Spring, Md., 1984, pp. 153-164.
19. D.P. Agrawal and J.S. Leu, "Dynamic Accessibility Testing and Path Length Optimization of Multistage Interconnection Networks," *IEEE Trans. Computers*, Mar. 1985, pp. 255-266.

Readers may write to Adams at RIACS, M/S 230-5, NASA Ames Research Center, Moffett Field, CA 94035.

Readers may write to Agrawal at Dept. of Electrical and Computer Engineering, Box 7911, North Carolina State University, Raleigh, NC 27695.

Readers may write to Siegel at the Supercomputing Research Center, 4380 Forbes Blvd., Lanham, MD 20706.



George B. Adams III is a research engineer with the Research Institute for Advanced Computer Science (RIACS). His research interests include computer architecture, parallel computing, interconnection network design, and modeling and analysis of computer systems. He holds a patent with H.J. Siegel for the extra stage cube interconnection network.

Adams received the PhD degree in 1984 and the MSEE degree in 1980 from the School of Electrical Engineering, Purdue University. His BSEE degree, granted in 1978, is from Virginia Polytechnic Institute and State University.

Dharma P. Agrawal is a professor of electrical and computer engineering at North Carolina State University, Raleigh. He is the group leader for the B_Hive multicomputer project at NCSU. His research interests include both software and hardware aspects of parallel and distributed processing, computer architecture, and fault tolerant computing.

Agrawal is a fellow of the IEEE. He was awarded a "Certificate of Appreciation" by the Computer Society. He is an editor of *Computer* and the *Journal on Parallel and Distributed Computing*. He is the author of the tutorial text *Advanced Computer Architecture* published by the Computer Society of the IEEE.

Howard J. Siegel leads the PASM parallel processing system project at the Supercomputing Research Center in Lanham, Maryland. From 1976 to 1987, he was a professor at Purdue. He has consulted, given tutorials, and published over 120 papers about parallel processing. He authored the book *Interconnection Networks for Large-Scale Parallel Processing* and was a distinguished visitor of the Computer Society.

Siegel is general chairman of the 15th Computer Architecture Symposium to be held in 1988. He is also associate editor of the *Journal of Parallel and Distributed Computing*. He received degrees from MIT and Princeton and is a member of the Eta Kappa Nu and Sigma Xi honorary societies.

The 7th
International
Conference on

**Distributed
Computing
Systems**

BERLIN (West), Germany,
21 – 25th September 1987
International Congress Center
Berlin

Preliminary Program

Sponsored by
The Computer Society of the IEEE
The Institute of Electrical and Electronic Engineers, Inc

in cooperation with
Hahn-Meitner-Institut Berlin
Gesellschaft fur Informatik e.V.

The Conference

The 7th ICDCS is dedicated to various aspects of distributed computing systems, one of the most challenging domains of computer science. The conference is a milestone for reviewing the state of the art at an international level and outlining future technical developments in distributed systems.

The 7th ICDCS will take place under the auspices of the Governing Mayor. The Senate of Berlin (West) will hold a reception in the famous Reichstag building. Historic Berlin, celebrating its 750th anniversary, will offer an exciting environment for further activities (festivals, exhibitions, entertainment).

Conference Program

Wednesday, Sept 23 – Friday, Sept 25:

The 7th ICDCS will embrace the theory, design and implementation of distributed systems. General sessions will include:

- decentralized computer architectures
- parallel processing
- concurrency control
- distributed algorithms
- distributed operating systems
- control techniques for distributed systems
- applications
- languages and programming support
- specification, verification and certification
- security in distributed systems
- resource allocation and scheduling
- fault tolerance
- protocol design
- distributed database systems
- performance analysis and evaluation
- local area networks
- testing and debugging techniques in distributed systems

Tutorials

Full day tutorials will be offered during the first two days of the 7th ICDCS:

Monday, Sept 21:

1. Security in distributed systems
   V. Gligor (University of Maryland, USA)
2. Highly available distributed systems
   F. Cristian (IBM, San José, USA)
3. MAP and TOP
   G. Workman (General Motors, Warren, USA)
4. Distributed operating systems
   A. Tripathi (University of Minnesota, USA)

Tuesday, Sept 22:

5. Real time systems and fault tolerance
   H. Kopetz (TU Wien, Austria)
6. ISDN and broadband-ISDN
   P. Kahl (FTZ Darmstadt, FRG)
7. Distributed databases
   S. Ceri (Politecnico di Milano, Italy)
8. Parallel processing networks and systems
   H. J. Siegel (Purdue University, USA)

Tutorial attendance is limited and will be allocated on a first-come, first-served basis, so please sign up as soon as possible.

Organization

The conference registration fee covers the admission to the technical sessions, one copy of the conference proceedings, refreshment, and lunches. The fee also includes admission to a banquet held at a lake side location. Tutorial registration covers the admission to the tutorial, tutorial text, refreshments, and lunch. Rooms are blocked in different hotel categories. Please make your **hotel reservation as soon as possible**, because in September 1987, Berlin will be a very busy place.

To participants travelling from the USA, the Travel Committee, Inc. offers special airfares on scheduled service airlines as well as hotel packages and post-conference tours. For reservation call: The Travel Committee, Inc. 1-800-638-6188 press 1 and then 4 for administration, ask for Sherry Bley, your personal travel consultant.

Post address: 110 Painters Mill Road, Owing Mills, MD 21117

Please send the registration form as soon as possible to the Conference Office CPO HANSER SERVICE GmbH. After receipt of registration form an Advance Program will be mailed. It will contain detailed information about the conference schedule, the supporting program and post-conference tours.

Conference Organization

General Chairman
H. Weber (FRG)

Honorary Chairman
K Zander (FRG)

E/A Chairman
R Popescu-Zeletin (FRG)

A/A/A Chairman
W C McDonald (USA)

E/A Program Chairman
G. Le Lann (F)

A/A/A Program Chairman
K H. (Kane) Kim (USA)

Decentralized Computer Architecture
Vice-Chair. W. D Carroll (USA)
H Aiso (Japan), D W. Mizell (USA), J. Lipovsky (USA), W. Giloi (FRG), J. S. Banino (F), I. M. Barron (U.K.)

Communication Architectures and Protocols
Vice-Chair: A Danthine (Belgium)
H Zimmermann (F), I. Chlamtac (Israel), A. Cantoni (Australia), E Manning (Canada), M. T. Liu (USA), W. W. Chu (USA)

Distributed Operating Systems
Vice-Chair: A van Tilborg (USA)
L Svobodova (CH), L Simoncini (I), J. Altaber (CH), J T Ellis (USA), T Lawrence (USA), E Lazowska (USA)

Applications
Vice-Chair: N Naffah (F)
S Schindler (FRG), M Sloman (U.K), P M Franca (Brazil), E. Shipley (USA), M Malek (USA)

Distributed Database Systems
Vice-Chair. H Garcia-Molina (USA)
G. Gardarin (F), H. J. Schneider (FRG), S. Ceri (I), P. B. Berra (USA), S. K. Rahimi (USA)

Software Engineering, Languages and Tools
Vice-Chair. M. Diaz (F)
T Maibaum (U K.), C. Vissers (NL), R. B. Grafton (USA), P. Lohr (FRG), Y Matsumoto (Japan), C. V. Ramamoorthy (USA), S. S. Yau (USA)

Reliability, Fault Tolerance and Security
Vice-Chair H. Kopetz (A)
J. C. Laprie (F), T Anderson (U K), A. Avizienis (USA), V. Gligor (USA), H. Hecht (USA)

Modelling and Performance Evaluation
Vice-Chair. T C Chen (Hongkong)
F Kamoun (Tunisia), O. Spaniol (FRG), L. Fratta (I), D. Ferrari (USA), T. L. Booth (USA)

Theory and Formalism
Vice-Chair: W. H. Kohler (USA)
P. Caspi (F), W. P. de Roever (NL), N. Francez (Israel), B. K. Bhargava (USA), R. Strong (USA)

International Liaison Committee

Chairman
Carl Davis (USA)

Vice Chairman
B Butscher (FRG)

Secretary
Peter Ng (USA)
M. C. Chruscicki (USA)
A. Scheller (FRG)

Members
R-J. Back (SF)
P. van Binst (B)
B. Carlson (S)
K. Chon (Korea)
L. O Hertzberger (NL)
P M. Franca (Brazil)
T. Kalin (YU)
H. Kerner (A)
L. Lenzini (I)
K.-S. Leong (Singapore)
K. Lindelien (N)
P. Linington (GB)
B. Mahon (L)
G. Muller (FRG)
A. Patel (IRL)
N. K. Pavlidis (GR)
J. Riera (E)
S Sedillot (F)
V. A Shaw (RSA)
M. Tokoro (J)
P. Villemoes (DK)
S. Zhongxiu (PR China)

Tutorial Chairmen
Earl E. Swartzlander (USA),
L Pouzin (F)

Awards and Special Publ.
Ch. R Vick (USA)

Secretary
R M. Yanney (USA)

Treasurer
V. Tschammer (FRG)

Local Arrangements
P. Egloff (FRG)
H. Troy Nagle (USA)

Important Dates:

Today	Complete and send the registration form attached
Sept 21–22, 1987	Tutorials
Sept 23–25, 1987	Conference

For the Advance Program and further information contact the conference organizer

R. Popescu-Zeletin
Hahn-Meitner-Institut
Glienicker Str. 100
D-1000 Berlin 39
West Germany

Phone
+ 49-30-80 09 2594
2541
Network address
COMPMAIL cmp5097

Registration Form ICDCS '87

Please return this form to:

CPO HANSER SERVICE GmbH
Schaumburgallee 12
D-1000 Berlin 19, West Germany
Phone + 49-30-305 31 31
Fax + 49-30-305 73 91
Telex 1 86 111 cpo d

Please fill in using a typewriter
or block capitals

☒ please mark with a cross

A. Participant

( ) Mrs. ( ) Mr.

Family Name: ____________________
First Name: ____________________
Title: ____________________
Institute/Company: ____________________
Street: ____________________
Postal Code, City: ____________________
Country: ____________________
Telephone: ____________________ Telex: ____________________

B. Registration Fees

Conference (excluding tutorials)	Payment received by **Aug 1, 1987**	Payment received after **Aug 1, 1987**
Member of IEEE and GI Membership No. ________	30 ☐ DM 500	31 ☐ DM 600
Non member	32 ☐ DM 625	33 ☐ DM 750
Student enclose proof of status	34 ☐ DM 100	34 ☐ DM 100

Registration fee for conference DM ____________

Tutorials (additional fee per tutorial per day)	Payment received by **Aug 1, 1987**	Payment received after **Aug 1, 1987**
Member of IEEE and GI Membership No. ________	No. of tutorials 35 ___ x DM 350	No. of tutorials 36 ___ x DM 450
Non member	37 ___ x DM 450	38 ___ x DM 565
Student enclose proof of status	39 ___ x DM 50	39 ___ x DM 50

Registration fee for tutorials DM ____________

I participate in the following tutorials:

	Monday, Sept 21				Tuesday, Sept 22			
Tutorial No.	1	2	3	4	5	6	7	8
Tick one per day								

Subtotal registration fees DM ____________

C. Hotel Reservation

Deadline: Aug 1, 1987

Please make the following reservation:

Arrival: ________ Departure: ________ No. of Nights: ____

Hotel category	Single room	Double room
L	☐ DM 180–240	☐ DM 215–290
A	☐ DM 140–180	☐ DM 175–215
B	☐ DM 100–120	☐ DM 145–175
C	☐ DM 55– 80	☐ DM 90–120

The prices shown are per room for one night and include service charge and tax. In categories L, A and B all rooms have private bath/WC or shower/WC. Category C also includes hotel pensions and small hotels, where some rooms only have hot and cold running water. Breakfast is included in the price at hotels in categories A–C. In hotels of category L there is an additional charge for breakfast.

Hotel rooms will be reserved only after receipt of the hotel deposit which includes DM 10 handling charge to cover the travel agents' administrative costs.

		No. of persons
Hotel deposit per person Categories L, A	100	______ x DM 250
Hotel deposit per person Categories B, C	100	______ x DM 130

Hotel deposit DM ________

Total sum
(registration fees + hotel deposit) **DM** ________

Important Note

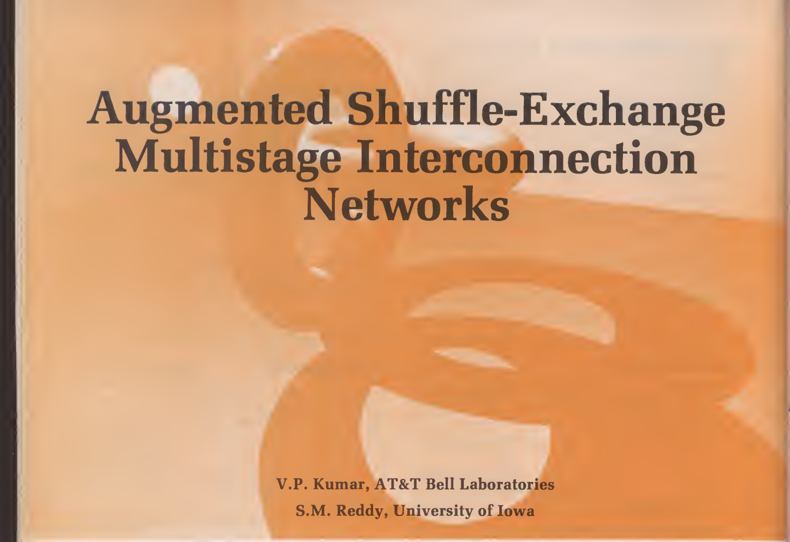
To guarantee your reservation, a check or a copy of remittance must be included with the conference registration form.
Bank account no. 10 5003 3333, Sparkasse der Stadt Berlin West, BLZ 100 500 00,
reference: CPO HANSER SERVICE w/ICDCS 1987.
Credit cards and personal cheques will not be accepted. Refunds of the registration fees (less DM 50 handling charge) will be made if the request is received in writing by Aug 1, 1987. No refund can be made in the event of cancellation of hotel reservation.

________________ ________________________
Date Signature

PAN AM. Official Carrier of the Conference

Augmented Shuffle-Exchange Multistage Interconnection Networks

V.P. Kumar, AT&T Bell Laboratories
S.M. Reddy, University of Iowa



Sculpture Simulation © Paul Brown/Digital Pictures 1983

A simple yet effective technique reduces the problems of poor performance and low reliability associated with unique-path multistage interconnection networks.

Multistage interconnection networks, or MINs, have long been studied for use in telephone switching and multiprocessor systems.[1] Since the early 70's several MINs have been proposed to meet the communication needs of multiprocessor systems in a cost-effective manner. These MINs are typically designed for $N = m^n$ inputs and N outputs and contain n stages of (N/m) crossbar switching elements of size $m \times m$. The switches in adjacent stages are connected in such a way that a path can be found from any input to any output. Such MINs have many properties that make them attractive for multiprocessor systems, including $O(N\log N)$ hardware cost as opposed to the $O(N^2)$ hardware cost of crossbar switches, the ability to provide up to N simultaneous connections, and $O(\log N)$ path lengths. Moreover, they can employ simple and distributed routing algorithms—algorithms that make a central routing controller unnecessary and also result in relatively low communication latency times.

MINs with $\log N$ stages also have two other important properties: a unique path exists from any input to any output, and distinct input/output paths may have common links. These lead to two chief disadvantages. First, an input/output connection may be blocked by a previously established connection (even if the outputs involved are distinct), thereby causing poor performance in a random-access environment. Second, the failure of even a single link or switch disconnects several input/output paths, leading to a lack of fault tolerance and low reliability. An example is provided by a MIN with 16 inputs and 16 outputs, built from 2×2 switching elements (Figure 1). The figure highlights all the input/output paths that use the link labeled l. It is clear that at any given time, at most one of these paths can be set up, and that the failure of link l disconnects all of these paths. The reduction in performance due to blocking and the decrease in reliability due to lack of fault tolerance become increasingly serious with the size of the network, because the num-

0018-9162/87/0600-0030$01.00©1987IEEE

ber of paths passing through a given link increases linearly with N.

Here, we discuss a simple yet effective technique to alleviate the problems posed by unique-path MINs, one that is applicable to a broad class of such networks—the shuffle-exchange[2,3] and *topologically equivalent* MINs.[4] A careful study of the structure of these MINs reveals that the addition of a few links per stage results in a rich collection of multiple paths between every network input/output pair. In setting up a connection (or, in a packet-switching environment, in routing a packet), multiple-path MINs allow an alternate path to be chosen if conflicts arise with other connections or if faults develop in the network. Thus, multiple-path MINs provide both better performance and higher reliability than unique-path ones.

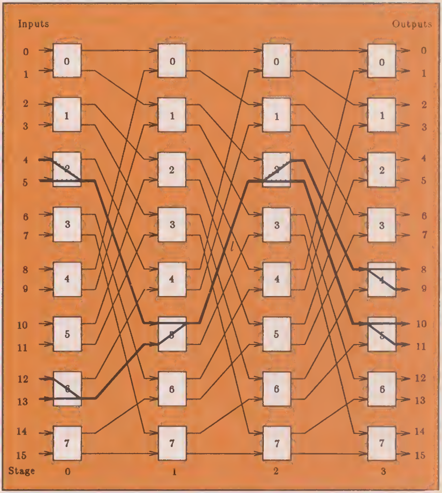


Figure 1. A unique-path shuffle-exchange MIN with 16 inputs and 16 outputs. The heavy lines indicate the paths using the link *l*.

Background and design objectives

In recent years, much research on multiple-path MINs has been reported.[5] Each network proposed aims to achieve a different objective. To understand the motivation behind the designs discussed here, let us briefly examine some quantitative and qualitative issues relating to multipath MINs.

Multiple paths may be an inherent part of the definition of a MIN, or they may be created by augmenting the topology of an existing MIN. In any case, their hardware cost is higher than that of unique-path MINs in terms of the number of switch stages, the number of switches per stage, and/or the size of the switching elements. These three factors contribute to what is usually called the *switch complexity* of a MIN. Another measure of the cost of a MIN is its *link complexity*, which depends on the number of intrastage links (if any), and the number of stages. Link complexity is an important measure because the implementation of a MIN is often I/O or pin limited at every level of integration. At the integrated-circuit level, the size of the switch (assuming that each IC contains one switch) is usually determined by the number of pins available and not by the complexity of the logic in the switch. At the printed-circuit-board level, the number of connectors available on a board is a critical factor in determining the number switches that can be placed on the board. At the wafer-scale-integration level, if a MIN with a large number of inputs and outputs were to be laid out on a single wafer,[6] most of the area would be occupied by the links and not by the switching elements. Of the two types of links in MINs, interstage links tend to be more expensive than intrastage ones. One of the main considerations in the design of the networks is keeping the switch and link complexities as low as possible.

A *fault model* captures the assumed effects of physical failures on the operation of a system. Three fault models are used for MINs: the stuck-at fault model, the link fault model, and the switch fault model. In the stuck-at fault model, a failure causes a switching element to remain in a particular state regardless of the control inputs given it, thus affecting its ability to set up proper connections. The affected switching element can be used to set up paths if the stuck-at state is also the required state. In the link fault model, a failure affects an individual link of a switching element, leaving the remaining part of the switch operational. In the switch fault model, the strongest of the three, a failure makes a switch totally unusable. The networks we discuss are designed to tolerate switch faults.

A key issue in multipath MINs is the manner in which rerouting or selection of alternate paths is achieved. The topology of a multipath MIN may allow rerouting to be done only at the source or at some fixed points in the network. In that case, a busy link, a faulty link, or a faulty switch encountered while setting up a path may necessitate backtracking to a stage at which a fork exists and then attempting to set up an alternate path from there. If the paths between every input/output pair in a multipath MIN have a fork at every

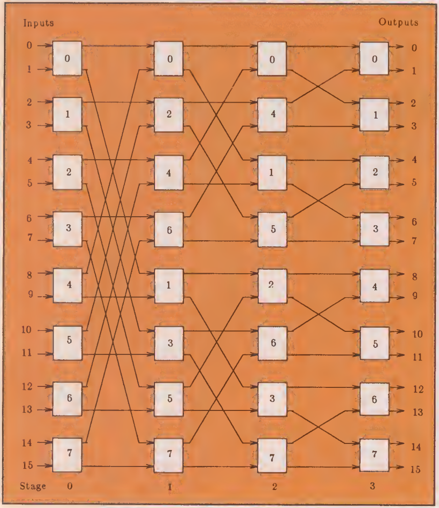


Figure 2. The shuffle-exchange MIN of Figure 1 redrawn to show the conjugate subsets of switches.

stage, then rerouting can be done on the fly without resorting to backtracking. However, backtracking MINs tend to have less hardware complexity than nonbacktracking ones. But backtracking MINs are inconvenient to implement, since they may require bidirectional paths and reverse queues. Also, if backtracking is to be avoided in these MINs for performance enhancement, it becomes necessary to have global fault information.

A *redundancy graph* offers a convenient way to study the properties of a multipath MIN, such as the number of faults tolerated or the type of rerouting possible. A redundancy graph depicts all the available paths between a source and a destination in a MIN. It consists of two distinguished nodes—the source S and the destination D—and the rest of the nodes correspond to the switches that lie along the paths between S and D. We make quite extensive use of redundancy graphs.

The rest of this article describes a technique for augmenting shuffle-exchange MINs and discusses the effectiveness of the resulting multipath networks. Our objective is to achieve

- single-switch fault tolerance in all stages, *including* the first stage and the final stage,
- the ability to reroute without backtracking,
- on-line repairability of faulty components, and
- low switch and link complexity.

The first three of these requirements are easily met by the MINs we described in an earlier paper,[7] which are also obtained by augmenting shuffle-exchange-type networks. But their complexity is relatively high. The augmentation technique we present here also achieves the fourth objective.

Shuffle-exchange MINs

Shuffle-exchange MINs can be built from many combinations of switch sizes,[8] and our augmentation schemes can be specified for the general case. For the sake of brevity and clarity, however, we will restrict our discussion to MINs built from 2×2 switches. Such a shuffle-exchange MIN is defined for $N = 2^n$ inputs and outputs and has n stages, each consisting of $N/2$ switches. The stages are numbered from 0 to $n-1$, and the switches in each stage are numbered from 0 to $N/2-1$. For convenience, the switch i in stage j is sometimes referred to as switch (i,j). The upper links (also called 0-links) of switch i are numbered $2i$, and the lower links (also called 1-links) are numbered $2i+1$. The interconnection pattern between adjacent stages is the perfect shuffle permutation. The output link i of stage j is connected to the input link $(2i + \lfloor 2i/N \rfloor) \bmod N$ of stage $j + 1$. The network shown in Figure 1 is, in fact, a shuffle-exchange MIN for $N = 16$.

Shuffle-exchange MINs allow routing to be done in a distributed fashion using the binary label of a destination as the routing tag. Let $d_0, d_1, \ldots, d_{n-1}$ be the binary address of the required destination. The switches along the path in stage i use the bit d_i and make a connection to the upper output link if the bit is 0 or to the lower output link if the bit is 1. It is the property of shuffle-exchange MINs that when the destination tag is used in this manner, a path is established to the proper destination. This can be verified by examining any of the paths shown in Figure 1. Notice that the routing tag for a destination stays the same regardless of the source to which the destination is to be connected.

Augmented shuffle-exchange networks

To achieve fault tolerance, we exploit the fact that there are subsets of switches in each stage which lie on paths leading to the same subset of destinations. For instance, any switch in stage 0 leads to any destination; all even-numbered switches in stage 1 lead to destinations 0 through $N/2-1$; and all odd-numbered switches in stage 1 lead to destinations $N/2$ through $N-1$. All the switches in a given stage which lead to the same subset of destinations comprise a *conjugate subset* of switches. The switches in each stage are partitioned into several conjugate subsets. The shuffle-exchange MIN of Figure 1 has been redrawn in Figure 2 to facilitate identification of the conjugate subsets of switches. One can easily verify that in stage i, two switches k and l belong to the same conjugate subset if $k \equiv l \bmod 2^i$. Further, the partial routing tag required to set up a connection to a reachable destination from either switch k or switch l is also the same.

In each conjugate subset of switches, there are several pairs of switches called *conjugate pairs of switches*.[9] The switches in such a pair are connected to the same switches in the next stage. For example, in Figure 2, the switches (0,0) and (4,0) form a conjugate pair since they are both connected to the switches (0,1) and (1,1). In general, the switches (i,j) and $(i+N/4,j)$, $0 \leq i < N/4$, $0 \leq j < n-1$, are both connected to the switches $(2i,j+1)$ and $(2i+1,j+1)$. Conjugate subsets and conjugate pairs of switches play a fundamental role in our augmentation scheme.

The scheme for creating multiple paths is based on connecting the switches belonging to a conjugate subset by using additional links to form loops. For this purpose, it is necessary to replace the 2×2 switches with 3×3 switches of the type shown in Figure 3a. The vertical links, which are used to connect switches in the same stage, are called *auxiliary links*. The network of Figure 2, augmented in this fashion, is shown in Figure 3b. This figure also shows how fault-tolerant routing can be accomplished. If a switch is not able to process a request for connection because of a faulty switch in the next stage or because of a busy link, it can route that request via its auxiliary output link to the next switch in the loop. The next switch can then make a connection to a different (nonfaulty) switch in the following stage. Suppose a connection from input link 9 to output link 4 in Figure 3b must be made. Under normal conditions, the path would pass through the switches (4,0), (0,1), (1,2), and (2,3). Now suppose that switch (1,2) is faulty. It can be bypassed using the path (4,0), (0,1), (2,1), (5,2), and (2,3). In fact, using an auxiliary link whenever a fault is encountered allows any source to be connected to any destination while tolerating any single faulty switch in any stage other than the initial stage and the final stage.

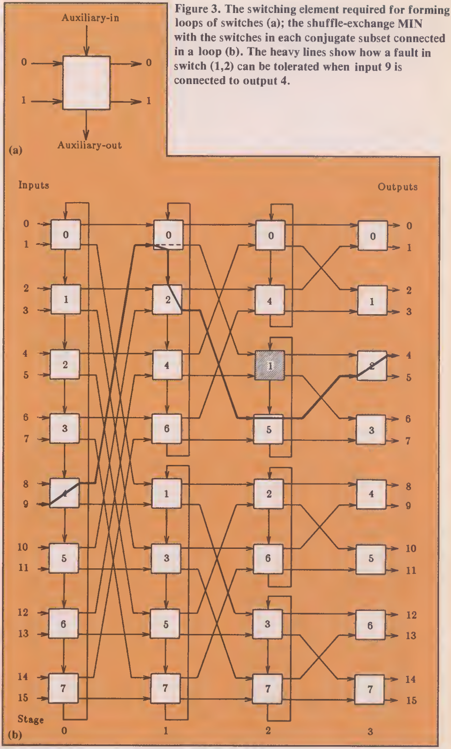


Figure 3. The switching element required for forming loops of switches (a); the shuffle-exchange MIN with the switches in each conjugate subset connected in a loop (b). The heavy lines show how a fault in switch (1,2) can be tolerated when input 9 is connected to output 4.

A switch failure in the initial stage, however, disconnects the sources attached to that switch from the rest of the network. Similarly, a switch failure in the final stage disconnects the destinations attached to that switch. To attain fault tolerance in these stages, each source and each destination *must* be provided with at least one additional I/O port, so they can be connected to at least two distinct switches. Schemes for connecting sources and destinations equipped with two ports each are shown in Figure 4. A 2×1 multiplexer is placed at each input link of stage 0, and each source is connected to two distinct multiplexers, as shown in Figure 4a. To provide additional connections to the destinations, the $N/2$ switches in the final stage of the MIN are replaced with N 1×2 demultiplexers. Each destination is directly connected to two demultiplexers, as shown in Figure 4b.

Next, if each loop in Figure 3 is divided into two loops, subject to the condition that no two switches in a given loop form a conjugate pair, then we obtain a network which has not only single fault tolerance but also certain properties that facilitate on-line repair and maintenance. Such a network is called an augmented shuffle-exchange network, or ASEN. An ASEN for $N = 16$ is shown in Figure 5a, and its redundancy graph is given in Figure 5b. This network is called ASEN-Max, since it has the maximum number of switches per loop within the constraints that the switches in a loop belong to the same conjugate subset and that no two switches in a loop form a conjugate pair. The number of switches in a loop in stage i of ASEN-Max is equal to 2^{n-2-i}. Note that in the final stage (stage $n-2$) there is only one switch per loop. Since there is no obvious advantage to connecting the auxiliary links of the same switch, they are left out in the final stage in Figure 5. In fact, 2×2 switches can be used in this stage.

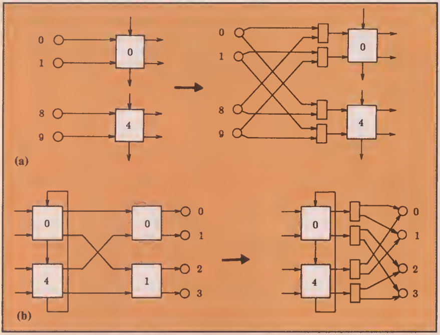


Figure 4. Schemes for making multiple connections for sources and destinations—addition of 2 × 1 multiplexers at the input stage (a); replacement of the switches in the final stage with 1 × 2 demultiplexers (b).

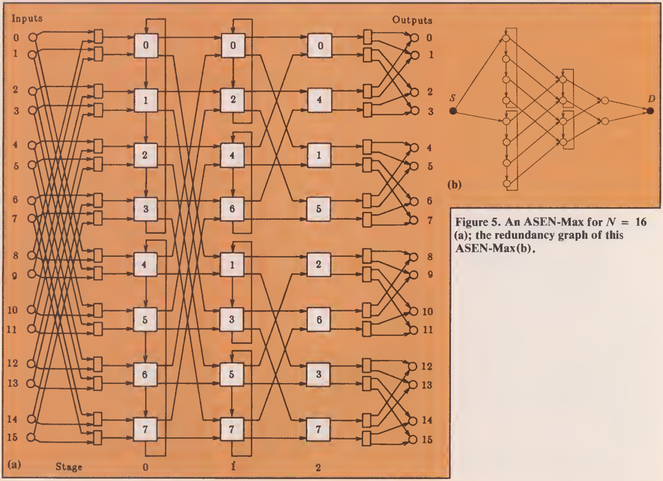


Figure 5. An ASEN-Max for $N = 16$ (a); the redundancy graph of this ASEN-Max(b).

It is possible to design ASENs with smaller loops than those of ASEN-Max and yet retain the property of single fault tolerance. For example, the ASEN in Figure 6 has a loop size of two in every stage except the last stage. This network is called ASEN-2. The loops in ASEN-2 are formed by switches that differ in the second most significant bit of their binary labels. In general, an ASEN in which the number of switches in a loop in stage i is equal to Min $(2^{n-2-i}, K)$ is called ASEN-K. A general principle to be followed in forming the loops is to ensure that if two switches are part of a loop, their conjugate switches will also be part of a (different) loop. All the loops in a given stage are required to contain an equal number of switches, thus restricting K to be a power of 2. This makes ASENs symmetric in the sense that the redundancy graph of a given source/destination pair is isomorphic to that of any other source/destination pair.

The redundancy graph for the ASEN-2 of Figure 6 and the redundancy graph for an ASEN-4 for $N = 32$ are shown in Figures 7a and 7b, respectively. Notice that the loops formed in all stages except the final stage are such that for every loop there exists another loop which is connected to the same set of switches in the next stage. Such pairs of loops are called *conjugate loops*. In the final stage, since a loop consists of only one switch, there are in effect conjugate pairs of switches connected to the same subset of destinations.

ASENs have $\log N - 1$ stages and $N/2$ switches of size 3×3 per stage, plus N 2×1 multiplexers and N 1×2 demultiplexers. The switch sizes and the overall switch complexity are both very low for these nonbacktracking and single-fault-tolerant MINs. Furthermore, ASENs have the same number of interstage links as unique-path MINs. All the additional links—except for those connecting the sources and the destinations—are of the intrastage type, which have a lower I/O cost.

In the physical design of ASENs, one can take advantage of their structure to reduce the I/O requirements of printed-

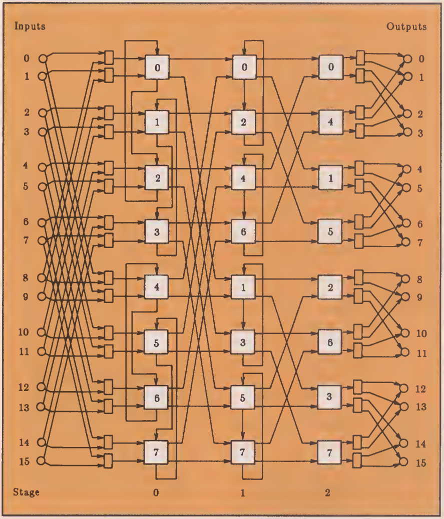


Figure 6. An ASEN-2 for N = 16.

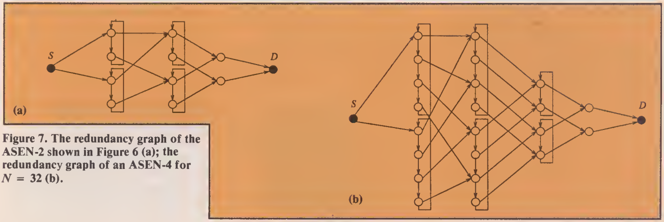


Figure 7. The redundancy graph of the ASEN-2 shown in Figure 6 (a); the redundancy graph of an ASEN-4 for N = 32 (b).

circuit boards or VLSI chips, by placing all the switches in a loop on the same board or chip. This confines all the auxiliary links to the same board or chip. If a loop consisting of L switches is placed on a board, the board must have enough connectors for $2L$ input links and $2L$ output links. Thus the number of switches one can connect in a loop is dependent on the I/O capacity of the boards.

A major advantage of implementing the loops as modules is that such implementation makes ASENs easier to maintain and repair. As we will show later, the fault tolerance of ASENs is not limited to single-switch faults but extends to faults affecting *all* the switches in a given loop. In other words, the removal of a loop as a whole from an ASEN need not disrupt the continued operation of the network. Thus, when a switch fault is identified, the loop containing the faulty switch can be removed from the network and a replacement loop inserted without interrupting the operation of the network. If more than one loop can be accommodated on a single printed-circuit board or a VLSI chip, then no two loops on a board or chip must form a conjugate pair of loops. It is economically attractive to have all the loops be of the same size so that only one type of replacement board is needed. This is an advantage of ASEN-2 networks over other ASENs.

Adaptive routing control

The reliability and performance improvement obtained from a multipath network depend upon how effectively the alternate paths available are used by the routing algorithm. One can use a backtracking routing algorithm that exhaustively searches for an available fault-free path. However, implementation of backtracking is relatively expensive in terms of hardware and backtracking can, in some situations, take an inordinately long time to set up connections. The nonbacktracking algorithm given below is easy to implement and performs quite well. This algorithm assumes that sources and switches have the ability to detect faults in the switches to which they are connected. Several techniques for detecting faults in multistage networks have been reported.[5] Such techniques usually require off-line application of test inputs or make the test results available only to the external resources attached to the network. If adaptive routing is to be implemented, with the switches making the routing decisions, then fault information is required by the individual switches at the time they process a request. In such situations, off-line diagnosis methods are not adequate. It is necessary to design switching elements capable of detecting faults in adjoining switching elements as they occur and concurrently with normal operation. One such design, which is based on *concurrent error detection* techniques, is discussed elsewhere[10] and assumed by the adaptive routing algorithm presented here.

The routing algorithm for ASENs

- For each source:
  Submit the request for connection to the primary switch, along with the binary label of the required destination. If the primary switch, or the multiplexer at the input of the primary switch, is faulty, then submit the request to the secondary switch. If the secondary switch is also faulty, drop the request.
- For each switch in stage $i (i < n - 2)$:
  (Requests for connection may arrive on any of the three input links.)
  For each request, use bit i of the associated destination tag and connect to the output link labeled correspondingly. If the required output link is busy or cannot be used because of a fault in the next stage, route the request via the auxiliary output link to the next switch in the loop. If the auxiliary link is also unusable because it is busy or because of a fault, then drop the request. A fault in a demultiplexer at the output of a switch in stage $n - 2$ is regarded as a fault in that switch.
- For each switch in stage $n - 2$:
  (Requests may arrive on any of two input links.)
  For each request, use bit $n - 2$ of the routing tag and route the request accordingly via the corresponding output link. If the required output link is busy, drop the request.
- For each demultiplexer at the output of stage $n - 2$:
  (May receive a maximum of one request.)
  If a request arrives, then use bit $n - 1$ of the routing tag and make a connection to the upper or the lower output link according to whether the routing bit is a 0 or a 1.
- For each destination:
  (Up to two requests may arrive.)
  Accept the requests.

This algorithm is easily incorporated in the hardware design of a switching element. The design of the switches can be uniform for all stages. At first, the switches in the final stage appear to be different in terms of the number of links and also because they are required to report faults in the demultiplexers at their outputs to the switches in the previous stage. However, designing all the switches in such a way that they report themselves to be faulty whenever their auxiliary link and at least one other output link lead to faulty switches, and then permanently typing the auxiliary output links of the switches used in the final stage to the faulty state, solves the uniformity problem.

Notice that the routing algorithm is independent of the number of switches in a loop. Thus, once the switching element is designed, loops of any number of switches can be formed to meet the performance and reliability requirements and the physical design restrictions. Also, in a circuit switching environment, routing along the loops does not result in any deadlocks—if a request traverses the entire loop and comes back to the first switch in the loop, it will be dropped unless it is possible to route it to the next stage. A simple way to prevent deadlocks from occurring in a packet-switched environment is to provide buffers at the output links to the next stage but not at the auxiliary output links.

Another problem peculiar to multiple-path MINs is that when they are used in a packet-switched environment, packets may arrive out of sequence. This can cause computational inconsistencies or, if countered by buffering, lead to a further increase in hardware and buffering delays. This problem can be resolved by using virtual circuit techniques or otherwise restricting the paths used when the proper sequence of packets has to be maintained. For example, the auxiliary links could be used only when a faulty switch is encountered but not when blocking occurs. Notice that this is not an issue in circuit-switched networks.

Fault tolerance and reliability of ASENs

The adaptive routing algorithm for ASENs allows two ways of routing a connection request in every stage except the final stage. If the destinations are assumed to be fault-free, it follows that all single-switch faults can be tolerated if this routing algorithm is used.

ASENs are, strictly speaking, single-fault tolerant. One can find combinations of two switches which, when simultaneously faulty, can disconnect a source from a destination. For instance, if both switches to which a source or a destination is connected become faulty, then that source or destination is disconnected from the rest of the network. However, if such critical combinations of switches are not present in a fault pattern, several multiple faults can be tolerated. We can characterize the faults that are tolerated and then derive probabilistic measures of the effect of faults, given that each switching element has a certain reliability.

A general criterion for evaluating the robustness of a MIN is that every member of a subset of sources must have paths to every member of a subset of destinations. The probability that the above criterion is satisfied is called *multiterminal reliability*. Two special cases of this criterion are of interest. The first case is that of the subsets of sources and destinations containing exactly one element each. This special case leads to a measure called *two-terminal reliability*, or simply *terminal reliability*, which is the probability that a given source/destination pair has at least one fault-free path between them, given that each switch has a certain reliability. (The reliability of a switch is the probability that it is fault-free.) The terminal reliability of a MIN can be evaluated using its redundancy graph. The other special case of the multiterminal reliability criterion is that of the subsets of sources and destinations containing all the sources and all the destinations, respectively. This special case leads to the assumption that the MIN is faulty (or disconnected) whenever all the paths are disconnected between some source/destination pair, and gives us the *reliability of the MIN*. If the reliability is time-variant one can obtain the mean time to failure, or MTTF, of the MIN, which is the time expected to elapse before some source is disconnected from some destination.

Recall that the nonbacktracking routing algorithm for ASENs does not explore all available paths. Thus, the relevant measure for one-to-one connections is not the probability of the existence of a fault-free path but the probability that the routing algorithm succeeds in finding one, in the absence of conflicts. The routing algorithm succeeds in making the connection if (a) at least one of the switches to which the source is connected is fault-free, (b) every subsequent switch can route either by the output link required by the routing tag or by its auxiliary output link, and (c) at least one switch visited in a loop can route the request to the next stage via the link required by the tag bit. The probability of success of the routing algorithm, called the effective terminal reliability, or ETR, is the same for all source/destination pairs if faults occur uniformly among the switches, since ASENs are symmetric networks.

The ETRs of ASEN-2, ASEN-4, and ASEN-Max—obtained analytically and checked by Monte Carlo simulation—are plotted in Figure 8. It is assumed that the probability of a switching element being fault-free is $p = 0.9$. Since a larger loop size creates more paths, one would expect an improvement in ETR with increasing loop size. Notice, however, that while the ETR of ASEN-4 is a little better than that of ASEN-2, it is practically the same as that of ASEN-Max. The reason is that although the ASENs with larger loop sizes have more paths, these paths are of longer lengths. The probability of occurrence of fault patterns that necessitate the use of the longer paths rapidly decreases, accounting for the saturation effect in the ETR. To show the improvement obtained over unique-path networks, we have also plotted the ETR of unique-path MINs (which is the same as their terminal reliability) in Figure 8.

To estimate the MTTF of ASENs, we must first characterize the faults that will or will not disconnect a source/destination pair. Consider the redundancy graph of any ASEN. Each of the conjugate loops in a stage is connected to both of the conjugate loops in the next stage. This guarantees that if faults affect switches in at most one loop in every pair of conjugate loops, and at most one switch in every pair of conjugate switches in the last stage, then no source can be disconnected from any destination. In fact, even the removal of an entire loop from every pair of conjugate loops will not disconnect the redundancy graph of any source/destination pair. This gives a lower bound on the maximum number of faults tolerated by ASENs: 50 percent of the total number of switches.

We note here that tolerance of faulty loops suggests an on-line repair procedure for ASENs. If a switch becomes faulty,

FREE!
Work better with your computer! Catalog of over 2,700 support products shows how.

Timesavers, spacesavers, moneysavers, sanity savers! Find them all each month in the Inmac catalog.

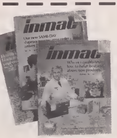
See flexible disks to furniture, print spoolers to plotter supplies, cables to cleaning kits—over 200 pages of quality products! And each is photographed and explained so clearly, even beginners learn how to buy wisely.

You get timesaving hints, technical help by phone, same-day shipment, free trials, 1-year-to-lifetime guarantees, and Inmac's 10 years of experience.

CALL TOLL-FREE TODAY!
1-800-547-5444
Or fill out & mail the coupon below.

Get the most from your computer... subscribe to the Inmac catalog!

inmac


Inmac, 2465 Augustine Drive, Santa Clara, CA 95054

Name
Company
Mail Stop/Suite/Dept./Bldg./Apt.#
Street Address
City State Zip

5125

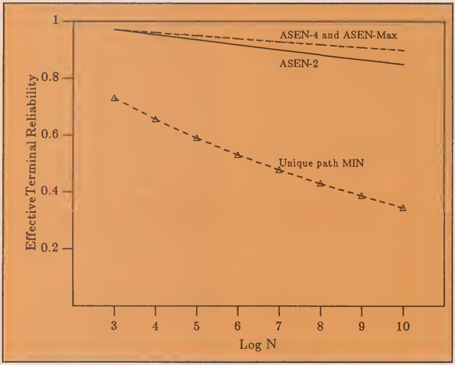


Figure 8. Effective terminal reliability of ASENs and the unique-path MIN.

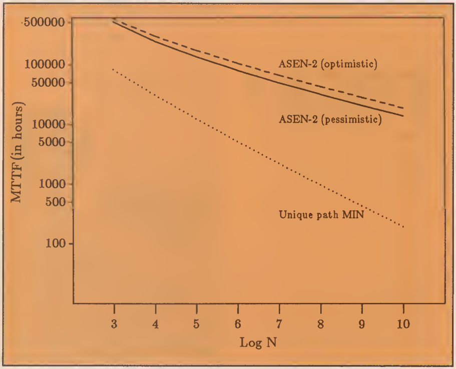


Figure 9. The MTTF of ASEN-2 and the unique-path MIN, at a switch failure rate of 10^{-6} per hour.

then the loop containing the faulty switch can be removed from the network and a replacement loop inserted. It is necessary to have a procedure for gracefully terminating the connections using the nonfaulty switches in the loop before removing the loop. However, the overall operation of the network need not be disrupted, and the network can continue to operate at a reduced level of performance while it is being repaired.

Using the loop fault tolerance property, we can get a lower bound for the MTTF of ASEN-2 networks. We evaluate this assuming that an ASEN-2 becomes faulty (i.e., that some source is disconnected from some destination) whenever more than one loop is affected by faults in a conjugate pair of loops (a pessimistic assumption). We evaluate the MTTF using a simple series-parallel reliability model: the two loops in a conjugate pair are in parallel and all the conjugate pairs of loops are in series.

Next, note that the pessimistic estimate of the MTTF of ASEN-2 is also a lower bound for ASEN-K, $K \geq 2$. This follows from the observation that if a multiple-switch-fault pattern does not cause the failure of ASEN-2, it cannot render an ASEN with larger loops faulty.

We can also obtain an optimistic estimate of the MTTF for ASEN-2. From its redundancy graph, note that in each stage there are two "gateway" switches that receive connections from the previous stage. These two form, in fact, a conjugate pair of switches. Also, switches in every conjugate pair in the network act as gateways to the stage they belong to, for some source/destination pair. Thus, if both the switches in a conjugate pair fail, then clearly some source is disconnected from some destination. If we assume that an ASEN-2 becomes faulty *only* when a conjugate pair of switches is present in a fault pattern (an optimistic assumption), then we get an upper bound for its MTTF. This case also has a series-parallel model, with the switches in a conjugate pair in parallel and all the conjugate pairs in series.

The optimistic and pessimistic estimates of the MTTF of ASEN-2 are plotted in Figure 9. It is assumed that the switches have a Poisson failure rate of $\lambda = 10^{-6}$ per hour. Notice that the curves corresponding to the two estimates of the MTTF of ASEN-2 are quite close and give a reasonably good estimate of the actual MTTF, which lies between the two curves. Also shown in this figure is the MTTF of unique-path MINs. Notice that the MTTF of ASEN-2 is at least an order of magnitude higher than that of unique-path MINs, for small sizes, and that the difference increases rapidly as the size of the networks increases. At $N = 512$, for example, the difference is two orders of magnitude. Adjusting the curve of unique-path MINs to reflect a slightly lower switch failure rate does not make a big difference: it only increases the MTTF figures by a small constant factor.

Performance of ASENs

As mentioned before, the performance of multiple-path MINs is better than that of unique-path MINs since alternate paths can be used to reduce the effect of blocking in a random-access environment. Here we discuss the performance of ASENs in a circuit-switched environment and show that the improvement is quite significant. The improvement in performance is due to the utilization of intrastage links, which results in longer path lengths than those in unique-path networks. Analytical results show, however, that the increase in path length is not excessive.

One measure used to quantify the circuit switching performance of a MIN is the *probability of acceptance*.[3] This is the probability that, in a random-access environment, a request submitted by a source is accepted by a destination without getting blocked by other requests or connections in the network. This probability is usually evaluated by assuming that all the sources simultaneously generate their requests for connection with a probability p and aim them at randomly chosen destinations, at the beginning of a cycle. These requests propagate through the network one stage at a time. When two or more requests arrive at a switch requiring the same output link, the requests that are serviced are chosen at random and the others are blocked and dropped. In the case of ASENs, a maximum of three requests can arrive at a switch. Up to two requests with the same routing tag bit can be serviced, by connecting one to the required output link to the next stage and the other to the auxiliary output link leading to the next switch in the loop. Destinations accept up to two requests per cycle, since they have two connections to the network. The probability of acceptance is defined as the ratio of the expected number of successful requests to the expected number of requests submitted by the sources.

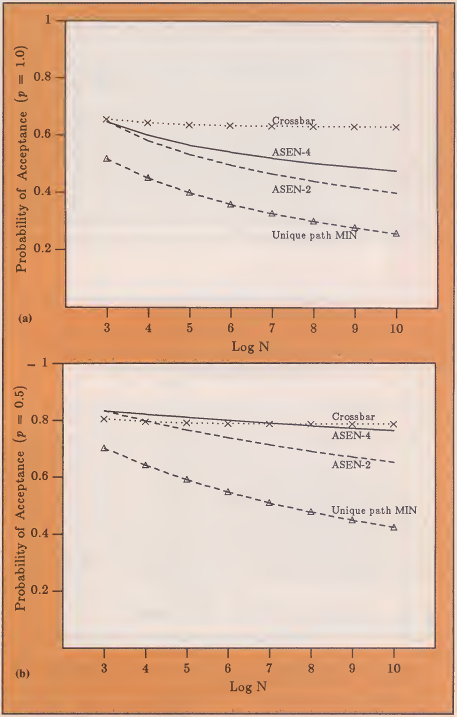


Figure 10. The probability of acceptance for ASENs, the unique-path MIN, and the crossbar switch—at a request generation probability of 1.0 (a) and 0.5 (b).

The results of the analysis of the probability of acceptance in ASENs are plotted in Figures 10a and 10b for request generation probabilities of 1.0 (capacity loading) and 0.5, respectively. The performance of ASEN-4 is noticeably better than that of ASEN-2 but very close to that of ASEN-Max (which has not been plotted, to keep the graph from becoming cluttered). The reason for the saturation effect, in spite of more paths in ASEN-Max, is that the probability XD? of succeeding by using paths of longer lengths becomes progressively lower. Figure 10 also compares the performance of ASENs with that of the crossbar switches and unique-path MINs. Notice that at the maximum request generate rate ASEN-4 performs worse than the crossbar but much better than the unique-path MIN. More interestingly, for small values of N, if the request generation rate is reasonably low (Figure 10b), the probability of acceptance is actually higher for ASENs than for the crossbar switch. The reasons are twofold: first, the destinations in ASENs are equipped with two input/output ports and can accept up to two requests instead of the one request crossbar switches can accept; second, the degree of blocking, which is already low for a small N and low traffic,

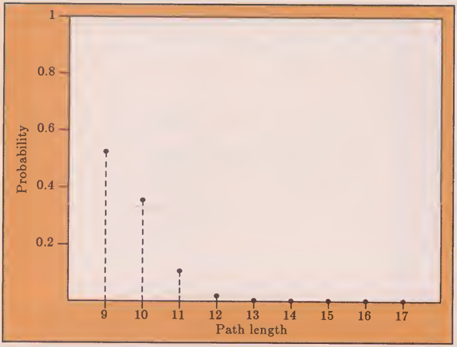


Figure 11. The probability distribution of the path length of a successful connection in ASEN-2, for $N = 1024$ and a request generation probability of 1.

is further decreased by the use of auxiliary links.

The price paid for higher probability of acceptance is a small increase in the expected path length. In Figure 11, the probability distribution of the path length of a successful connection, under capacity loading, in an ASEN-2 for $N = 1024$ is given. As can be seen, the probability mass is concentrated in the smaller values of the path length. For example, the probability of the path length being less than or equal to 11 is 0.98.

Our techniques for augmenting shuffle-exchange-type MINs result in the following advantages:

- single-switch fault tolerance,
- alternate routing without backtracking,
- a relatively small increase in link and switch complexity,
- ease of repair and maintenance, and
- significantly improved reliability and performance.

The networks resulting from the augmentation procedure—ASENs—feature links between switches belonging to the same stage, forming loops of switches. The number of switches in a loop can be varied up to a maximum. Although the number of alternate paths increases with increasing loop sizes, analysis and simulation of reliability and performance show that there is a saturation effect after a maximum loop size of about four.

By combining the advantages of shuffle-exchange unique-path MINs with the features mentioned above, ASENs become very attractive for multiprocessors as well as for voice and data switching applications. To test the ideas presented here, we are now implementing a version of ASENs at AT&T Bell Laboratories. In the initial testbed, we intend to use an ASEN in place of an S/Net bus[11] in an existing multiprocessor system. □

References

Additional references can be obtained from the authors.

1. T. Feng, "A Survey of Interconnection Networks," *Computer*, Dec. 1981, pp. 12-27.
2. D.H. Lawrie, "Access and Alignment of Data in an Array Processor," *IEEE Trans. Computers*, Dec. 1975, pp. 1145-1155.
3. J.H. Patel, "Performance of Processor-Memory Interconnections for Multiprocessors," *IEEE Trans. Computers*, Oct. 1981, pp. 771-780.
4. C. Wu and T. Feng, "On a Class of Multistage Interconnection Networks," *IEEE Trans. Computers*, Aug. 1980, pp. 696-702.
5. G.B. Adams, D.P. Agrawal, and H.J. Siegel, "A Survey and Comparison of Fault-Tolerant Multistage Interconnection Networks," *Computer*, this issue.
6. D.S. Wise, "Compact Layout of Banyan/FFT Networks," *Proc. CMU Conf. VLSI Systems and Computations*, Computer Science Press, Rockville, Md., 1981, pp. 186-195.
7. S.M. Reddy and V.P. Kumar, "On Fault-Tolerant Multistage Interconnection Networks," *Proc. 1984 Int'l Conf. on Parallel Processing*, Computer Society Press, Washington, D.C., 1984, pp. 155-164.
8. L.N. Bhuyan and D.P. Agrawal, "Design and Performance of Generalized Interconnection Networks," *IEEE Trans. Computers*, Dec. 1983, pp. 1081-1090.
9. V.P. Kumar and S.M. Reddy, "Design and Analysis of Fault-Tolerant Multistage Interconnection Networks with Low Link Complexity," *Proc. 12th Ann. Int'l Symp. Computer Architecture*, Computer Society Press, Washington, D.C., 1985, pp. 376-386.
10. V.P. Kumar, "On Highly Reliable, High Performance Multistage Interconnection Networks," PhD thesis, University of Iowa, Dec. 1985.
11. S.R. Ahuja, "S/NET: A High Speed Interconnect for Multiple Computer," *IEEE J. Selected Areas in Communication*, Dec. 1983, pp. 751-756.

Vijay P. Kumar is with the System Architectures Research Department at AT&T Bell Laboratories in Homdel, New Jersey. His interests include fault-tolerant and distributed computing, communication architectures, and performance and reliability evaluation.

Kumar received a BS in electronics and communications engineering from Osmania University, Hyderabad, India, and an MS and PhD in electrical engineering from the University of Iowa.

Sudhakar M. Reddy is chairman of the Department of Electrical and Computer Engineering at the University of Iowa, where he has been a faculty member since 1968. His interests include fault-tolerant computing, design for testability, VLSI circuit design, and distributed computing.

Reddy received a BSc in physics and a BE in electrical and communications engineering from Osmania University, Hyderabad, India, and an ME in electrical and communication engineering from the Indian Institute of Science, Bangalore. He also holds a PhD in electrical engineering from the University of Iowa. He is a fellow of the IEEE.

Readers may write to Vijay Kumar at Room 4G-608, AT&T Bell Laboratories, Crawfords Corner Road, Holmdel, NJ 07733.

CVPR'88

CALL FOR PAPERS

IEEE Computer Society Conference on Computer Vision and Pattern Recognition

University of Michigan
Ann Arbor, Michigan
June 5-9, 1988

GENERAL CHAIR
Professor Ramesh Jain
Computer Science Department
University of Michigan
Ann Arbor, MI 48104

PROGRAM CHAIR
Professor Larry S. Davis
Center for Automation Research
University of Maryland
College Park, MD 20742

PROGRAM COMMITTEE

Narendra Ahuja	Ramakant Nevatia
Ruzena Bajcsy	Edward M. Riseman
Oliver D. Faugeras	Jorge Sanz
Martin A. Fischler	Steven Shafer
Eric Grimson	Brian G. Schunck
Anil Jain	William Thompson

SUBMISSION OF PAPERS

Each paper should have a cover sheet with the title, authors' names, primary authors' address and telephone number, index terms containing at least one of the above topics, and the type of paper (long or short). Papers will be considered only for the type indicated. The body of the paper must contain an abstract of about 250 words. The length of the papers should be appropriate for the final length of the selected type (the number of double spaced typed pages will be reduced by a factor of about 3 when typed on the model paper). Deadline for long and short papers will be November 11, 1987. Authors will be notified of acceptance by January 7, 1988. Final papers will be required by The Computer Society of the IEEE on February 29, 1988. Send four (4) copies of papers to:

Professor Larry S. Davis
Center for Automation Research
University of Maryland
College Park, MD 20742

REVIEW OF PAPERS

In order to maintain the quality of the conference, all papers will be reviewed by two members of the program committee.

Conference topics include:

- Image processing
- 3-D representation and recognition
- Motion
- Stereo
- Visual navigation
- Shape from (shading, texture, . . .)
- Vision systems
- Vision architectures
- Pattern recognition
- Applications of computer vision
- Knowledge-based vision

The program will consist of submitted and invited papers. Invited papers will provide an overview of research in particular areas. Submitted papers may be long or short. Long papers (8 proceedings pages, 30 minutes presentation) are intended for completed, high quality work. Short papers (6 proceedings pages, 20 minutes presentation) are for continuing work with partial results and for ideas that can be presented concisely. The time and space allowed for short papers should permit complete presentation of results.

FOR FURTHER INFORMATION

Mail to: CVPR'88
c/o The Computer Society
1730 Massachusetts Ave., N.W.
Washington, DC 20036-1903

Please send further information as it becomes available.

Name ______________________

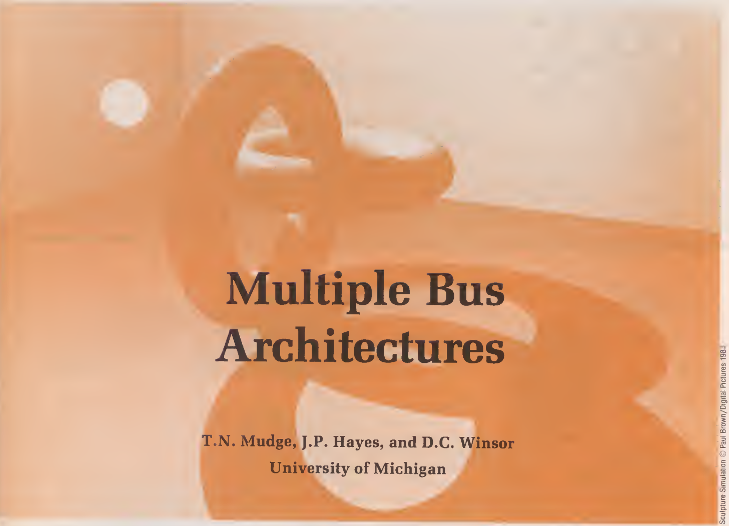
Affiliation ______________________

Address ______________________

City, State, Zip ______________________

THE COMPUTER SOCIETY OF THE IEEE

IEEE
THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC



Sculpture Simulation © Paul Brown/Digital Pictures 1984

Multiple Bus Architectures

T.N. Mudge, J.P. Hayes, and D.C. Winsor
University of Michigan

By replacing the single shared bus of conventional multiprocessor architectures by a set of buses, the number of processors can be increased to a few hundred.

A recent study[1] noted that for shared memory multiprocessors the single system bus typically used to connect the processors to the memory "... is by far the most limiting resource, and system performance can be increased considerably by increasing the capacity of the bus." One way of increasing the bus capacity, and also the system's reliability and fault tolerance, is to increase the number of buses. In this article we discuss using multiple buses to provide high-bandwidth connections between the processors and the shared memory, thereby allowing the construction of larger and more powerful systems than currently possible.

The architecture we have in mind is shown in Figure 1. It contains N processors, $P_1, P_2, \ldots, P_N$, each having its own private cache, and all connected to a shared memory by B buses, $B_1, B_2, \ldots, B_B$. The shared memory consists of M interleaved banks, $M_1, M_2, \ldots, M_M$ to allow simultaneous memory requests concurrent access to the shared memory. This avoids the loss in performance that occurs if those accesses must be serialized, which is the case with only one memory bank.

Each processor is connected to every bus and so is each memory bank. When a processor needs to access a particular bank, it has B buses from which to choose. Thus, each processor-memory pair is connected by several redundant paths, which implies that the failure of one or more paths can, in principle, be tolerated at the cost of some degradation in system performance.

In a multiple bus system several processors may attempt to access the shared memory simultaneously. To deal with this, a policy must be implemented that allocates the available buses to the processors making requests to memory. In particular, the policy must deal with the case when the number of processors exceeds B. For performance reasons this allocation must be carried out by hardware arbiters which, as we shall see, add significantly to the complexity of the multiple bus interconnection network.

0018-9162/87/0600-0042$01.00©1987IEEE

Techniques for interconnecting multiple processors and, in particular, connecting multiple processors to a shared memory, have been the object of considerable study during the past decade. Perhaps the most studied type of interconnection network is the multistage network. Many have been proposed, and we refer the reader to Siegel[2] for a thorough discussion of the subject.

The rationale for most multistage networks is that they have fewer components than a crossbar switch while retaining many of the desirable connectivity properties of the crossbar. This reduction in interconnection complexity is important because the systems for which multistage networks are targeted are envisioned to have hundreds or thousands of processors. A few commercial machines based on multistage interconnection networks have been produced, such as the BBN Butterfly. These networks have also been proposed for a number of experimental machines, such as the Illinois Cedar project, the IBM RP3, and the Purdue PASM.

In contrast, the shared memory multiprocessors emerging as commercial products have, with few exceptions, avoided multistage networks as a means of interconnecting the processors to the shared memory. Instead they employ interconnection architectures based on conventional buses. Representative examples are the Encore Multimax and Sequent Balance 8000, which use a single shared bus.[1] One consequence of this evolutionary approach to interconnection design is that the commercial systems have been limited to a few dozen processors rather than thousands of processors. The multiple-bus interconnection network retains the well-understood features of the single bus, but may allow construction of larger systems that can compete with the multistage systems in computing power.

A typical shared system bus contains 50 to 100 wires and is physically realized by printed lines on a circuit board or by discrete (backplane) wiring. Additional costs are incurred by interface electronics, such as line drivers, receivers, and connectors. Concern about wiring and interface costs is probably a major reason why multiple buses have seen little use so far in multiprocessor designs.

In this article we explore some of the practicalities of extending a single shared bus system that can support a few dozen processors to a multiple bus system that can support two or three times as many processors. To illustrate the issues involved, we first describe the operation of a hypothetical multiple bus system in more detail.

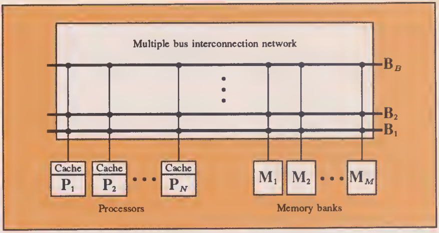


Figure 1. Multiple-bus multiprocessor.

System operation

The importance of the caches in the multiple-bus system of Figure 1 is that they reduce the number of processor-memory requests (the bus traffic) that need to refer back to the shared memory, allowing the construction of more powerful systems without increasing the number of buses. This is important because, as noted above, the buses and their associated interface and control logic form a significant fraction of the system cost. Unlike the uniprocessor case, the cache of a multiprocessor does not have to be constructed from very fast (and expensive) memory, as its primary function is to reduce the traffic with the shared memory, and thus to avoid the potential performance degradation associated with having to wait for a shared resource. The extent to which references have to be made to shared memory depends a lot on the caching algorithm used. A key aspect of this algorithm is the policy for updating the shared memory when a request is addressed to the cache. In this article we assume a simple *copy back* or *write back* updating technique, where an altered cache block is copied to shared memory only when the block is deallocated from the cache.

Copy back divides the references to shared memory into two categories: those due to cache misses and those due to the copy back process. A cache miss occurs when a processor generates a read or write request for data or instructions that cannot be satisfied by the cache. A cache block containing the requested word is read from the shared memory or one of the other caches into the cache where the miss occurred. This replaces a block currently in the cache.

The case where a cache is the source of a requested block occurs when the most recently modified version of the block is in some cache rather than in the shared memory; we refer to this as a cross-cache hit. Caches check for cross-cache hits addressed to themselves by monitoring memory addresses (snooping) on the system buses. Congestion can occur at a cache when it receives several simultaneous cross-cache hits from the buses.

Replaced cache blocks can be discarded (overwritten) if not modified while in the cache. On the other hand, if modified they must be copied back to the shared memory. A processor need not wait for a copy back operation to be completed if a write buffer is used at the interface to the buses.

Ensuring the consistency (coherence) of the data stored in multiple caches can get quite complicated. We refer the reader to Archibald and Baer[1] for a detailed discussion of this issue.

Since both read misses and copy back operations require a cache block to be transferred on a bus, we assume that the data path width of the buses matches the block size of the cache. In the case of reads, the memory address information will precede the data transfer by the memory access time. To avoid idling the data path for this period, the address path and data path sections of a bus ought to run asynchronously.

Two sources of conflict due to memory requests are present in the system of Figure 1. First, more than one request can be made to the same memory module. Second, available bus capacity may be insufficient to accommodate all the requests. Correspondingly, the allocation of a bus to a processor that makes a memory request requires a two-stage process as follows:

(1) Memory conflicts are resolved first by M 1-of-N arbiters, one per memory bank. Each 1-of-N arbiter selects one request from up to N requests to get access to the memory bank.

(2) Memory requests selected by the memory arbiters are then allocated a bus by a B-of-M arbiter. The B-of-M arbiter selects up to B requests from the M memory arbiters.

The assumption that the address and data paths operate asynchronously allows arbitration to overlap with data transfers.

At first glance it may seem that there should be one bus per memory bank. This obviates the need for the second level of arbitration, yielding a double saving—we can drop the B-of-M arbiter and use a faster bus cycle. However, the optimal configuration may require significantly fewer buses than memory banks. To illustrate this, we next develop a simple performance model of multiple-bus systems.

Performance modeling

If we model the requests made by a processor for memory as a sequence of independent Bernoulli trials,* we can develop expressions for an "average" number of memory requests made by the N processors at the start of each memory cycle and the bandwidth of the interleaved memory. Let r be the probability that an arbitrary processor P_i requests access to shared memory at the start of a memory cycle (this is the Bernoulli trial), then the expected number of requests for shared memory is given by $N\,r$.

Some requests will always be blocked due to the two types of conflict mentioned earlier, no matter how large we make B and M. The bandwidth BW of the memory subsystem composed of the B buses and the M memory banks is defined as the expected number of busy memory banks, which is also the expected number of successful memory accesses. The presence of conflicts means that not all of the $N\,r$ expected memory requests make successful memory accesses, therefore $BW < N\,r$.

The requests for shared memory as modeled above can be reads resulting from cache misses; copy backs; or resubmissions of misses and copies blocked in earlier cycles due to conflicts with other processors accessing the same memory bank. When two or more processors attempt to access memory, an arbiter is invoked to give access to just one of them.

If we assume that the memory banks interleave on the low-order bits of the cache block address and that the instruction fetches and data accesses intermix, then some empirical evidence suggests that requests to all memory banks are equally likely.[3] In other words, the probability that the request from processor P_i is for a particular memory bank M_j is r/M, independent of i or j. Following Mudge et al.,[4] we split the development of an expression for BW into two steps corresponding to the two levels of arbitration.

(1) *Memory arbitration*. The probability that there are no requests from P_i to M_j is $1 - r/M$, and therefore the probability that none of the processors request M_j at the start of a memory cycle is $(1 - r/M)^N$. Let E_j be the event that there is at least one request for M_j; in other words, that the 1-of-N arbiter for M_j outputs a request. Then

$$Pr[E_j] = 1 - \left(1 - \frac{r}{M}\right)^N \qquad (1)$$

for any j. If the events E_j, $j = 1, \ldots, M$, are assumed independent and there are always a sufficient number of buses, i.e., $B \geq M$, then the expected number of busy memory banks is

$$BW_s = \sum_{j=1}^{M} Pr[E_j] = M\left[1 - \left(1 - \frac{r}{M}\right)^N\right]$$

where the subscript S denotes sufficient buses. In the case of large N this expression has an approximate lower bound given by

$$BW_s \approx M\left(1 - e^{-\frac{Nr}{M}}\right) \qquad (2)$$

(In the example given later, this approximation results in an error of less than 0.2 percent.) This expression, or variations on it, has appeared in the literature on numerous occasions. One of the earliest derivations came from Strecker.[5] In specific cases, Equation (2) can be evaluated by estimating r from the miss ratio, the cache block size, the ratio of memory cycle time to processor cycle time α, and the caching algorithm for the processors.

In the most general case of interest, $B < M$. This leads to the next step of the analysis.

(2) *Bus arbitration*. The assumption that the E_j's are independent allows us to express $f(i)$, the probability that exactly i of the memory arbiters output a request at the start of a memory cycle, as follows:

$$f(i) = \binom{M}{i} Pr[E_j]^i (1 - Pr[E_j])^{M-i}$$

In the case where $i \leq B$, there are sufficient buses to handle the memory requests and the B-of-M arbiter does not have to block any requests. In the case where $i > B$, all the B buses are in use and the B-of-M arbiter blocks $i - B$ of the requests. With these two cases in mind, we can write the expression for the expected number of memory banks in use as

$$BW = \sum_{i=1}^{B} i f(i) + \sum_{i=B+1}^{M} B f(i) \qquad (3)$$

where the two terms on the righthand side correspond to the two cases, $i \leq B$ and $i > B$. It is easy to show that when $B = M$, $BW = BW_S$; however, in general $N\,r > BW_S > BW$, when $r > 0$. These inequalities correspond to conflicts that cause memory requests to be blocked during memory arbitration ($N\,r > BW_S$) and then during bus arbitration ($BW_S > BW$). Goyal and Agerwala[6] first derived Equation (3) for BW, about the same time that Mudge et al.[4] extended it to the partial bus case. Das and Bhuyan[7] later used it in a reliability study.

Mudge et al.[8] observed that the derivation leading to Equation (3) relies on two assumptions: temporal independence and spatial independence. Temporal independence requires that successive memory requests by a processor be independent, which is clearly not valid for resubmitted blocked requests. Spatial independence corresponds to independent E_j's, an assumption that also has limited validity.

Consider a system with two processors, two buses, and two memory banks. The condition for spatial independence is

---

*Repeated (statistically) independent trials are called Bernoulli trials if there exist only two possible outcomes for each trial and their probabilities remain the same throughout. An example of this is a sequence of coin flips.

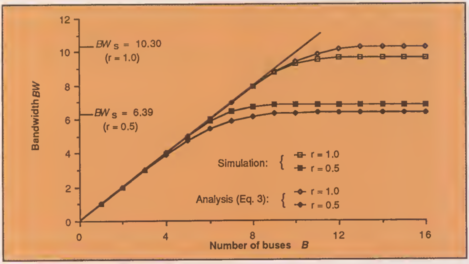


Figure 2. Graphs of *BW* versus *B* for $N = M = 16$.

$$Pr[E_1E_2] = Pr[E_1]Pr[E_2] \tag{4}$$

However, $Pr[E_1] = Pr[E_2] = r(1 - (r/4))$ but $Pr[E_1E_2] = r^2/2$. In other words, Equation (4) is not satisfied for $1 \geq r > 0$. The inaccuracy introduced by the two independence assumptions is illustrated in Figure 2, which compares the values of BW calculated from Equation (3) to those obtained by a simulation in which neither temporal nor spatial independence was assumed. In view of the simplifying assumptions in the underlying model, deviation of the computed results from the simulation data is quite small. Note that when the number of buses B is less than BW_S, the buses are the limiting factor, i.e., $BW \approx B$. When the number of buses exceeds BW_S, then the bandwidth can approach its maximum (bus-sufficient) value.

Other models address the accuracy of the independence assumptions. For example, Valero et al.[9] and Bhuyan[10] present several equivalent models in which only temporal independence is assumed. Although more complicated to develop, their models are only slightly more accurate than the model discussed above leading to Equation (3). On the other hand, iterative improvement techniques allow for temporal dependence and yield somewhat more accurate results when applied to Equation (3) and the other models cited above.[8] Towsley[11] developed a model that gives bandwidth predictions generally within one percent of simulation, at the cost of considerable computation. Using semi-Markov processes, Mudge and Al-Sadoun[12] obtained a model that extends to multicycle memory accesses. This model is useful when modeling systems where the block size exceeds the bus data path.

Finally, another class of multiple bus memory interference models, based on the work of Marsan and Gerla, employ traditional Markovian queueing network techniques.[13,14] Their distinguishing feature is that memory access time is treated as an exponentially distributed random variable. However, the fixed access times incorporated into the earlier models are the norm in real memories.

The remainder of our discussion illustrates the use of the performance model leading to Equation (3) and its implications for the design of the multiple bus subsystem.

Design example

Consider the design of the multiple bus architecture of Figure 1, where M and B are to be determined. Suppose that from performance studies of the candidate processors on the anticipated workload, it has been determined that $N = 64$ processors are needed, and that requests for memory from these processors should be successful 90 percent of the time. We make the following three assumptions:

(1) The processor cycle time is half the cycle time of the shared memory, i.e., $\alpha = 2.0$.

(2) The cache has a capacity of 64 kilobytes with a block size of 16 bytes. For such caches the observed miss ratio m has been in the neighborhood of 0.01.[15]

(3) Misses are frequently followed by copy back operations. Statistics published in Smith[15] indicate that the resulting double bus cycles occur about half of the time.

From these three assumptions we obtain an estimate for r of $1.5m\alpha = 0.03$. This leads to an expected number of $Nr = 1.92$ requests for shared memory at the start of each memory cycle.

As we saw in the earlier analysis, it is impossible to satisfy all requests because of conflicts due to memory reference patterns and insufficient bus capacity. The mismatch between memory bandwidth and memory requests can be conveniently characterized by p, defined by

$$p = \frac{BW}{Nr} \tag{5}$$

and corresponds to the probability of a successful request, which we require to be 90 percent. The memory subsystem design problem can now be viewed as choosing values of B and M, for a given N and r so that p reaches a minimum acceptable value. We solve this in two steps. First, we estimate lower bounds on B and M assuming a sufficient number of buses are available. These bounds are used to restrict the search space when solving Equation (3) for values of B and M that satisfy the constraint on p.

It follows from Equation (2) and Equation (5) that the probability of a successful request p_S if there are sufficient buses is given by

$$p_s = \frac{BW_s}{Nr} \tag{6}$$

This is inherent in the memory request patterns and depends only on M. Clearly,

Table 1. Success probability p_S for representative values of M assuming sufficient buses.

M	p_s
4	0.80
8	0.89
16	0.94
32	0.97

Table 2. Success probability p for representative values of M and B.

M	$B=2$	$B=4$	$B=8$
16	0.73	0.92	0.94
32	0.74	0.95	0.97
64	0.74	0.96	0.99

$p_S > p$, as (insufficient) buses can only make things worse. Our design specification calls for processor requests for memory to be successful 90 percent of the time, i.e., $p > 0.9$. The quantity p_S must also satisfy this bound, therefore we can use Equation (2) and Equation (6) to estimate M, the number of memory banks required. Table 1 below shows values of p_S for selected values of M, which are powers of 2 for efficient address interleaving.

From Table 1 we can see that at least 16 memory banks are needed to satisfy the requirement $p_S > p > 0.9$ and to make M a power of 2. Also, since $Nr = 1.92$, we can conclude that at least two buses are needed. Therefore, we can restrict our attention to values of $M \geq 16$ and $B \geq 2$, when using Equation (3) to obtain combinations of B and M that meet the bounds on p. This leads to Table 2, from which we see that $B = 4$ and $M = 16$ yields $p = 0.92$. Hence the system configuration with 4 buses and 16 memory banks is the desired solution to our example problem.

Lang et al.[16] were among the first to recognize in their study of multiple bus systems that the number of buses could be significantly less than the number of memory banks. The savings in the number of buses can be substantial, as we can see from our example, but come at the cost of requiring a B-of-M arbiter. In the next section we will look at this cost in detail.

Arbiter design

As we have seen, a general multiple bus system calls for two types of arbiters: 1-of-N arbiters to select among processors and a B-of-M arbiter to allocate buses to those processors that successfully obtained access to memory.

1-of-N arbiters. If multiple processors require exclusive use of a shared memory bank and access it on an asynchronous basis, conflicts may occur. These conflicts can be resolved by a 1-of-N arbiter. The typical signaling convention between the processors and the arbiter is as follows: Each processor P_i has a request line R_i and a grant line G_i. Processor P_i requests a memory access by activating R_i, and the arbiter indicates the allocation of the requested memory bank to P_i by activating G_i.

Several designs for 1-of-N arbiters have been published.[17] In general, these designs can be grouped into three categories: fixed priority schemes, rings, and trees.

Fixed priority arbiters are relatively simple and fast, but they have the disadvantage that they are not fair: lower priority processors can be forced to wait indefinitely if higher priority processors keep the memory busy.

A ring-structured arbiter gives priority to the processors on a rotating basis, with the lowest priority given to the processor that most recently used the memory bank requested. This has the advantage of being fair, because it guarantees that all processors will access memory in a finite amount of time, but the arbitration time grows linearly with the number of processors.

A tree-structured 1-of-N arbiter is generally a binary tree of depth $\log_2 N$ constructed from 1-of-2 arbiter modules (see Figure 3). Each 1-of-2 arbiter module in the tree has two request input lines, each with a corresponding grant output line, and a cascaded request output and a cascaded grant input for connection to the next arbitration stage. Tree-structured arbiters are faster than ring arbiters because the arbitration time grows as $O(\log_2 N)$ instead of $O(N)$. Fairness can be assured by placing a flip-flop in each 1-of-2 arbiter, toggled automatically to alternate priorities when the arbiter receives simultaneous requests.

Pearce, Field, and Little[17] give an implementation of a 1-of-2 arbiter module constructed from 12 gates. The delay from the request inputs to the cascaded request output is 2Δ, where Δ denotes the nominal gate delay, and the delay from the cascaded grant input to the grant outputs is Δ. Thus, the total delay for a 1-of-N arbiter tree is $3\Delta\log_2 N$. Therefore, to construct a 1-of-64 arbiter for our example, we need a six-level tree. This tree will contain 63 1-of-2 arbiters, for a total of 756 gates. The corresponding total delay imposed by the arbiter will be 18Δ.

B-of-M arbiters. Lang and Valero[18] gave detailed implementations of B-of-M arbiters. The basic arbiter consists of an iterative ring of M arbiter modules A_1, A_2, ..., A_M that compute the bus assignments, and a state register to store the arbiter state after each arbitration cycle (see Figure 4). The storage of the state is necessary to make the arbiter fair by taking into account previous bus assignments. After each arbitration cycle, the highest priority is given to the module immediately following the last one serviced—a standard round-robin policy.

An arbitration cycle starts with all the buses available. The state register identifies the highest priority arbiter module, A_i, by asserting signal e_i to that module. Arbitration begins with this module and proceeds around the ring from left to right. At each arbiter module, the R_i input is examined to see if the corresponding memory bank M_i is requesting a bus. If a request is present and a bus is available, the address of the first available bus is placed on the BA_i output and the G_i signal is asserted. BA_i is also passed to the next module, to indicate the highest numbered bus that has been assigned. If a module does not grant a bus, its BA_i output is equal to its BA_{i-1} input. If a module does grant a bus, its BA_i output is set to $BA_i + 1$. When $BA_i = B$ all the buses have been used and the assignment process stops.

The highest priority module A_i, as indicated by the e_i signal, ignores its BA_{i-1} input and begins bus assignment with the first bus by setting $BA_i = 1$. Each module's C_{i-1} input is a signal from the previous module that indicates that the previous module has completed its bus assignment. Arbitration proceeds sequentially through the modules until all of the buses have been assigned, or all the requests have been satisfied. The last module to assign a bus asserts its s_i signal. This is recorded in the state register, which uses it to select the next e_i output so that the next arbitration cycle will begin with the module immediately after the one that assigned the last bus.

Turning to the performance of B-of-M arbiters, we observe that the simple iterative design of Figure 4 must have a delay proportional to M, the number of arbiter modules. By combining g of these modules into a single module (the "lookahead" design of Lang and Valero[18]), the delay is reduced by a factor of g.

If the enlarged modules are implemented by PLAs with a delay of 3Δ, the resulting delay of the arbiter is about $(3M/g)\Delta$. For our example, where $M = 16$ and $g = 4$, the arbiter delay is about 12Δ. This allows the complete arbitration process for this example to be implemented with delay 30Δ, 18Δ for the 1-of-N arbiter, and 12Δ for the B-of-M arbiter. Since arbitration can be overlapped with bus accesses, the memory bus cycle time must be at least 30Δ.

If the lookahead design approach of Lang and Valero[18] is followed, the

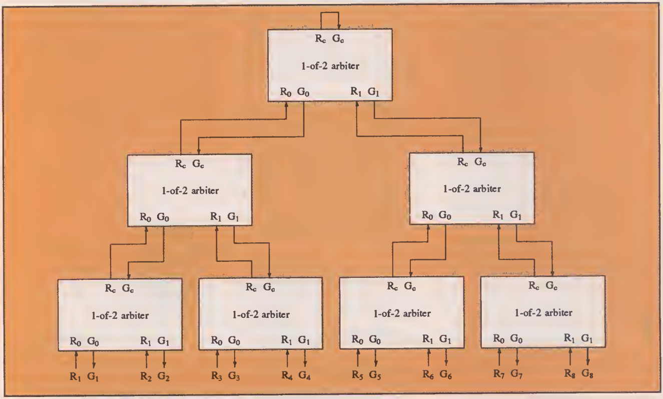


Figure 3. 1-of-8 arbiter constructed from a tree of 1-of-2 arbiters.

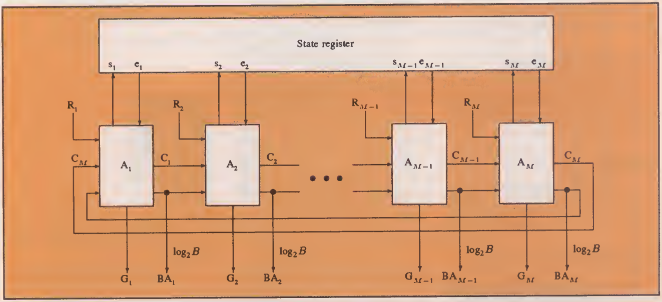


Figure 4. Iterative design for a B-of-M arbiter.

arbitration time of B-of-M arbiters grows at a rate greater than $O(\log_2 M)$ but less than $O(\log_2^2 M)$. Thus the delay of the B-of-M arbiter could become the dominant performance limitation for large M. This, however, is not a problem in our example with $M = 16$.

Like the arbiters, the bus conductors make a significant contribution to the overall system cost. Some rough calculations show that our four-bus example does not require an excessive number of (backplane) conductors. To transfer a cache block in one bus cycle requires 16×8 data conductors; combining these with a 32-bit address and 10 control signals gives a total of 170 conductors. Thus four buses will require 680 conductors. These are easily accommodated by two standard 384-pin 10-inch PCB edge connectors, which leave adequate additional conductors for power and ground. If four processors can be placed on a card, and the shared memory plus the arbiters can be placed on four cards, the system will fit on 20 cards, exclusive of the I/O subsystem. Placing the connectors on the backplane at 5/8-inch centers results in a backplane that measures about 2 feet × 1 foot.

Packaging technology is likely to set a limit of about four buses for the near future. If we assume the four buses can be time multiplexed and that larger caches than those proposed in the example are possible, then a four-bus system could support several hundred processors. If optical backplanes become a reality, the number of buses that could be supported will greatly increase, allowing future shared-memory multiple bus systems with thousands of processors. □

Acknowledgments

This work was supported in part by Army Research Office grant no. DAAG29-84-K-0070 and by the Office of Naval Research under contract N00014 85 K 0531.

References

1. J. Archibald and J.-L. Baer, "Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model," *ACM Trans. Computer Systems*, Nov. 1986, pp. 273-298.
2. H.J. Siegel, *Interconnection Networks for Large-Scale Parallel Processing: Theory and Case Studies*, Lexington Books, Lexington, Mass., 1985.
3. C.H. Hoogendoorn, "A General Model for Memory Interference in Multiprocessors," *IEEE Trans. Computers*, Oct. 1977, pp. 998-1005.
4. T.N. Mudge et al., "Analysis of Multiple-Bus Interconnection Networks," *Proc. 1984 Int'l Conf. Parallel Processing*, Aug. 1984, pp. 228-232.
5. W.D. Strecker, *Analysis of the Instruction Execution Rate in Certain Computer Structures*, PhD thesis, Carnegie Mellon University, Pittsburgh, Penn., 1970.
6. A. Goyal and T. Agerwala, "Performance Analysis of Future Shared Storage Systems," *IBM J. Research and Development*, Jan. 1984, pp. 95-98.
7. C. Das and L. Bhuyan, "Bandwidth Availability of Multiple-Bus Multiprocessors," *IEEE Trans. Computers*, Oct. 1985, pp. 918-926.
8. T.N. Mudge et al., "Analysis of Multiple-Bus Interconnection Networks," *J. Parallel and Distributed Computing*, Sept. 1986, pp. 328-343.
9. M. Valero et al., "A Performance Evaluation of the Multiple Bus Network for Multiprocessor Systems," *Proc. ACM Conf. Performance Evaluation*, 1983, pp. 200-206.
10. L.N. Bhuyan, "A Combinatorial Analysis of Multibus Multiprocessors," *Proc. 1984 Int'l Conf. Parallel Processing*, Aug. 1984, pp. 225-227.
11. D. Towsley, "Approximate Models of Multiple Bus Microprocessor Systems," *IEEE Trans. Computers*, Mar. 1986, pp. 220-227.
12. T.N. Mudge and H.B. Al-Sadoun, "A Semi-Markov Model for the Performance of Multiple-Bus Systems," *IEEE Trans. Computers*, Oct. 1985, pp. 934-942.
13. M.A. Marsan et al., "Modeling Bus Contention and Memory Interference in a Multiprocessor System," *IEEE Trans. Computers*, Jan. 1983, pp. 60-72.
14. I.H. Önyüksel and K.B. Irani, "A Markovian Queueing Network Model for Performance Evaluation of Bus-Deficient Multiprocessor Systems," *Proc. 1983 Int'l Conf. Parallel Processing*, Aug. 1983, pp. 437-439.
15. A.J. Smith, "Cache Evaluation and the Impact of Workload Choice," *Proc. IEEE 12th Ann. Int'l Symp. Computer Architecture*, June 1985, pp. 64-73.
16. T. Lang, M. Valero, and I. Alegre, "Bandwidth of Crossbar and Multiple-Bus Connections for Multiprocessors," *IEEE Trans. Computers*, Dec. 1982, pp. 1227-1233.
17. R.C. Pearce, J.A. Field, and W.D. Little, "Asynchronous Arbiter Module," *IEEE Trans. Computers*, Sept. 1975, pp. 931-932.
18. T. Lang and M. Valero, "M-users B-servers Arbiter for Multiple-Buses Multiprocessors," *Microprocessing and Microprogramming*, 1982, pp. 11-18.

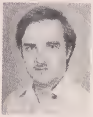
Readers may write to the authors at the Advanced Computer Architecture Laboratory, Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109.

Trevor N. Mudge is an associate professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. His research interests include computer architecture, programming languages, and computer vision.

Mudge received a BSc in cybernetics from the University of Reading, England, in 1969, and an MS and PhD in computer science from the University of Illinois in 1973 and 1977, respectively. He is a senior member of the IEEE.

John P. Hayes is a professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. He is also director of the department's Advanced Computer Architecture Laboratory. His research interests include computer architecture, VLSI design, digital system testing, and switching theory.

Hayes received the BE degree from the National University of Ireland in 1965, and the MS and PhD degrees from the University of Illinois in 1967 and 1970, all in electrical engineering. He is a fellow of the IEEE.



Donald C. Winsor is pursuing a PhD in electrical engineering at the University of Michigan. His research interests include computer architecture and VLSI design.

Winsor received the BSE degree in electrical engineering in 1981, the BSE degree in computer engineering in 1982, and the MSE degree in electrical engineering in 1983, all from the University of Michigan.

FOR SOFTWARE AND HARDWARE ENGINEERS

Digital has it now.

Breathing Room

DECwest is located in the great Northwest, where you'll find plenty of room to stretch out and grow.

At DECwest, you'll also find plenty of professional space. It's the ideal environment for the engineer obsessed with creating the most astounding technical marvels possible. And, if you're the type of person we're looking for, that's exactly what you'll do.

We're Digital's Western Engineering Group, and we're currently developing a system that will change the way computers are used, not just 5 years out, but a decade from now. This kind of project requires an atmosphere of technical creativity and individual initiative. That's what DECwest is about. As a result, we're looking beyond the obvious and the accepted. We're finding ways to take technical marvels and apply them to real world needs.

To find out more about DECwest, read on. If you have expertise in any of the following areas, send us your resume.

Then take a nice deep breath.

Software:

FILE SYSTEM EXTENSIONS

We're looking for individuals with experience in either or both of the following areas:

Record Management:

- Record access techniques such as read ahead, write behind, etc.
- Multi-buffering techniques
- Knowledge of VAX-RMS internals a plus
- Experience directly manipulating files with OS primitives a plus
- Internal knowledge of VMS* file system a plus

File Systems:

- Disk and/or tape storage organization techniques and algorithms
- Operating system development experience a plus
- Kernel-level programming a plus

DISTRIBUTED OS PRODUCT RESEARCH

You should have knowledge of the following:

- Client/server communications models
- Integrated high performance distributed services
- Distributed system security issues

BASE SYSTEMS

You'll need knowledge of the following:

- VMS operating systems internals and data structures
- The VMS lock manager
- Security kernels
- Multi-processor synchronization techniques

I/O SUBSYSTEMS

You'll need experience in at least two of the following:

- Device drivers
- High-performance I/O subsystems
- Real-time programming
- Data-communications drivers (e.g. ethernet drivers)
- State-of-the-art optimizing mass storage controllers

Basic background necessary for software positions includes 5 or more years of experience, a B.S. in Computer Science or equivalent and programming in high-level languages.

Hardware:

PROJECT LEADER — DIAGNOSTIC ENGINEERING

In this position you'll be responsible for the development of a total fault management strategy for the next generation computing system. This includes all facets of the problem, from system software error logging to hardware repair level diagnostics, including state-of-the-art symptom directed diagnosis.

Minimum requirements include: recent hands-on experience in a related field, 5 years of programming on VAX or a similar system, and a good working knowledge of computer systems hardware. Knowledge of VAX/VMS, Pascal, or C would be a definite plus.

DESIGN ENGINEER

You'll be part of the team chartered with designing and implementing a high-performance ECL computer system. You'll help design a major portion of the high-speed system.

Requires one year experience in hardware design and/or specific experience in one or more of the following areas: high-speed logic design, floating point logic design, CPU, memory, and/or peripheral systems design. Experience with simulation and timing verifications CAD tools is helpful but not required.

If qualified, please send your resume, indicating position you are applying for, to: **L. Taylor, Manager, Dept. 0601-7133, DECwest Engineering Group, Digital Equipment Corporation, 14475 NE 24th, Bellevue, WA 98007.**

We are an affirmative action employer.

*Trademark of Digital Equipment Corporation

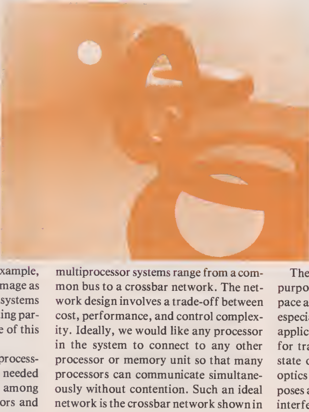
digital™

Optical Crossbar Networks

A.A. Sawchuk, B.K. Jenkins, and C.S. Raghavendra, University of Southern California

A. Varma, IBM T. J. Watson Research Center

Optical techniques solve fundamental problems in data communications. This article describes several design concepts for optical matrix-vector implementations of dynamic crossbar networks employing bulk (3-D) optical information processing architectures.



Sculpture Simulation © Paul Brown/Digital Pictures 1983.

With advances in technology and the declining cost of computer hardware, it is now feasible to design computer architectures consisting of large numbers of processors executing programs concurrently. Most parallel computing architectures can be generally classified into SIMD and MIMD systems.[1] SIMD processors simultaneously and synchronously execute the same instructions on different data streams, while MIMD processors are not necessarily synchronous and execute different instructions on different data streams. In image processing, for example, low level tasks generally have an image as input and image as output. SIMD systems well suit these tasks, and most existing parallel image processing systems are of this type.

An important part of a parallel processing system is the interconnection needed for the transfer of information among processors, and between processors and memories. In such systems several processors may need data from memories or other processors simultaneously; therefore, the network should be capable of providing fast parallel access to data.

The interconnections designed for use in multiprocessor systems range from a common bus to a crossbar network. The network design involves a trade-off between cost, performance, and control complexity. Ideally, we would like any processor in the system to connect to any other processor or memory unit so that many processors can communicate simultaneously without contention. Such an ideal network is the crossbar network shown in Figure 1.

A normal crossbar can be defined as a switching network allowing arbitrary one-to-one interconnection of N inputs to N outputs without contention. A *generalized crossbar* allows some inputs to have no output connection and other inputs to connect to two or more outputs (broadcasting). A generalized crossbar is the most desirable type of interconnection network to use in parallel processing because it is *rearrangeable* (any permutation of input ports to output ports can be realized) and *non-blocking* (connections between input and output ports can be established independently without disturbing other connections). Unfortunately, as the number N of processing elements grows, the hardware costs of an electronic crossbar become quite high.

The use of optics for communication purposes has increased at an enormous pace and received considerable attention, especially in local network and data bus applications. Fiber optic communication for transfer of data has reached a high state of development. The attributes of optics that make it useful for such purposes are its relative freedom from mutual interference and the high temporal bandwidth of some systems, which may permit the multiplexing of data from a large number of sources over a single optical channel at rates higher than those available from electronics. The parallel nature of optics may result in system complexity

0018-9162/87/0600-0050$01.00©1987IEEE

only weakly dependent on the size of the system.

There exist many different optical devices providing switching capability.[2-4] Generally, with existing technology, the time needed to switch optical signal lines is slower than the time needed to switch electronic signal lines, although the signal bandwidth of optical signal lines is much higher. We expect that the switching time of optical signal lines will decrease significantly with rapidly advancing developments in this technology.

In this article we present design concepts for several bulk (3-D) optical implementations of dynamic crossbar networks implemented as matrix-vector multipliers, and discuss the tradeoffs in their performance.

Crossbar design issues

We summarize several important issues in the design of a crossbar network as follows:

(1) *Reconfiguration time*: The total time needed to completely change the settings of all switches in the network, thus establishing a completely different interconnection pattern. The reconfiguration time could be shorter than, equal to, or larger than the duration of one data bit, provided that a protocol is established to ensure that no messages are lost within the network.

(2) *Synchronous or asynchronous operation*: Refers to the synchronism in the timing of changes of internal switches in an interconnection network and in the timing of the data bits. In an asynchronous interconnection network, the internal switching of a data line may occur independently of the data and switches on other lines. Synchronous networks require careful timing of data and internal changes of switch states according to a global clock; buffers are generally needed at the input and output to enforce this synchronism.

(3) *Signal bandwidth*: Defined as the bandwidth or bit rate per line achievable for a given data error rate.

(4) *Total data throughput*: The product of the number of lines N switched in the network and the signal bandwidth per line.

(5) *Data word size*: The number of bits that can be switched in parallel.

Until recently, large electronic crossbar networks were bulky and expensive to construct. The advent of very large scale integration (VLSI) permitted the integration of hardware for thousands of switches into a single chip. However, the number of pins on a VLSI chip cannot exceed a few hundred, which restricts the size of the largest crossbar that can be integrated into a single VLSI chip.[2,4] Larger crossbars can be realized only by partitioning them into smaller crossbars, each implemented using a single chip. Thus, a full crossbar of size $N \times N \times B$ bits can be implemented out of modules of size $n \times n \times b$ bits. This requires $N^2/n^2 \cdot B/b$ modules. As an example, the implementation of a crossbar of size 1024 × 1024 × 1-bit out of 32 × 32 × 1-bit switch modules requires 1024 modules. Problems with such realizations include difficulty of expansion, and a number of implementation problems including power and clock distribution, control, and reliability. Moreover, several delays become significant as the system size increases. If wire delays are assumed significant, the delay goes up as $O(N)$; the control decoder also contributes at least $O(\log_2 N)$ delay. Thus, a large amount of parallelism is lost because of the physical limitations of the technology.

Some practical implementations

IBM has constructed a few electronic crossbar switch units as part of experimental special-purpose parallel processor systems for logic simulation. The Los Gatos Logic Simulation Machine (LSM) employs a 64 × 64 unidirectional crossbar switch to interconnect 64 processing elements.[5] The processors operate synchronously and so does the switch. The switching system is controlled by a set of switch-select memories. Each processor has a switch-select memory associated with it, a 1024-word × 6-bit memory. The 6-bit address uniquely identifies the processor from which data is needed in the current cycle. The switch-select memories are initially loaded by a host computer. Since the configuration is preprogrammed, no conflicts can ever arise. The switch also has broadcast capability—any number of processors can access the output from any one processor simultaneously.

The Yorktown Simulation Engine (YSE),[6] a sequel to the Logic Simulation Machine, uses a 256 × 256 × 3-bit wide crossbar switch to interconnect 256 logic processing elements. This switch is constructed out of 256 individual 256 × 1 multiplexers. The switching system takes up an enormous amount of space—a total of 8 boards, each of size 14 × 18 inches. Due to this enormous size, we cannot reasonably expect the system to operate reliably over long periods of time. Moreover, the possibility of any further expansion is very difficult.

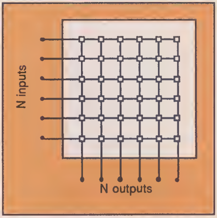


Figure 1. The crossbar network.

Broomell and Heath[7] discussed the implementation of a large electronic crossbar system using special-purpose VLSI chips. One system includes VLSI chips that are 16 × 15 crossbars, with a 480-output system conceivable. It is an asynchronous implementation in terms of both control and transfer of data. We can expect a cycle time of at least one microsecond with an input clocking rate of 10 megahertz. The system also has broadcast capability. By arranging the switching planes perpendicular to the plane of both the processor cards and the memory cards, expansion in terms of word size is easy. However, addition of more processors or memories requires changes in the card sizes. Moreover, if ever such a large system is built, it remains to be seen how reliably it can operate over long periods of time.

Optical spatial light modulators and switching devices

Many appealing characteristics of optical signals are potentially valuable for applications in computing.

First, optical systems are inherently multidimensional and parallel. For example, a simple imaging system can be thought of as a device that transfers one 2-D plane of data points to another 2-D plane in parallel.

Second, optical systems have fewer planar or quasiplanar constraints as in VLSI[2,3], and have a very large bandwidth

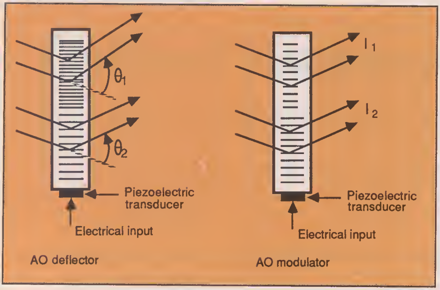


Figure 2. Acousto-optic (AO) cells.

in both the spatial and temporal domains.

Third, optical systems can be constructed with large space-bandwidth or time-bandwidth products. These parameters are essentially the product of the maximum space (or time) bandwidth of a system times the maximum physical spatial size (or time duration) over which a signal can be maintained. The space (or time) bandwidth product is one measure of the number of independent channels available for computing.

Fourth, optical signals in separate channels can propagate through each other with minimal interaction, which may allow a high density of interconnections with low crosstalk. Optical technologies may also avoid capacitive loading and other scaling problems associated with electronic VLSI,[2] reducing the problems associated with power and clock distribution and wire delays in an electronic implementation.

Spatial light modulators (SLMs) are an essential part of any optical information processing or computing system. SLMs store, modify, and amplify optical fields containing signals. Many different materials, designs, and technologies have been employed in SLMs.[3,4,8] SLMs can be optically or electronically controlled; different versions can process optical signals in 1-D, 2-D, or 3-D formats.

In the past, most SLMs were optimized for analog signal processing, thus linear input-output characteristics for SLMs in these applications over a wide dynamic range were desirable. More recently, many groups have investigated optical logic and switching technologies. Some of the desirable characteristics are a sharp threshold in the device input-output curve; low switching time, delay time, and power consumption; and input and output signals at the same wavelength and at standardized levels (for ease of cascading). Another desirable feature is to densely pack logic and switching devices into a large array to minimize their physical size.

Optical logic or switching systems may also operate in a hybrid arrangement (such as electronic control of optical switches, or vice versa). Although this hybrid operation is generally undesirable for the construction of an all-optical numerical processor or CPU, hybrid systems may have a role in interconnection networks.

One example of an electro-optic SLM that can implement switching and logic functions optically is the Hughes liquid crystal light valve.[3,8] The Hughes LCLV is an optical image transducer that accepts a low-intensity input 2-D image and converts it to an output image read out with illumination from another source. The LCLV consists of a sandwich of a cadmium sulfide (CdS) photoconducting layer and a thin cell containing liquid crystal (LC) material. Various other layers provide insulation and electrodes to operate the device. A bias voltage provides an electric field perpendicular to the layers. An input image can be made to spatially modulate the input impedance of the photoconductor, thus altering the electric field impressed across the LC layer and varying its electro-optical properties.

In other versions of the LCLV, application of varying electric fields on the device can be achieved by driving it with an electrical signal. The electro-optical properties of the LC material cause the LCLV to rotate or modulate the polarization of incident readout light. This polarization is converted to an intensity modulation with polarizing filters.

The Hughes LCLV can be operated so it has a nonlinear characteristic approximating a threshold function. An array of optical inputs or controls to the LCLV can control readout by an array of optical outputs; thus the device can function as an optical gate array or optically-controlled optical switch array. The device currently operates at approximately television frame rates (30-millisecond response for all gates) with spatial resolution of approximately 20 cycles per millimeter. The device has an aperture size of up to 50 millimeters by 50 millimeters, so the potential space-bandwidth product, or number of resolvable elements, is of the order of 10^6.

Another example of SLM technology, the Sight-Mod magneto-optic spatial light modulator[4] from Semetex Corp., is a commercial version of a similar device developed by Litton Data Systems. To manufacture the device, a thin magnetic garnet film is epitaxially deposited on a transparent, nonmagnetic garnet substrate. Semiconductor photolithographic techniques are used to etch the film into a two-dimensional array of mesas (individual cells or pixels) and to deposit electrical X-Y drive lines for matrix-addressed current switching of the mesa magnetization. The polarization of light passing through any of the cells is rotated by the Faraday effect. Only two polarization states are possible for each cell, thus the device is inherently binary in its optical modulation capability. The Sight-Mod device is placed between a polarizer and analyzer to convert the polarization rotation into transmission modulation. Another important feature is that the device has memory. That is, it retains the magnetic states of the cells without any refreshing needed. For image display applications, the individual magnetic mesas are normally 100-micron squares, although different sizes are possible. A change of state of the magnetic mesas can be initiated with a current pulse on the associated X-Y drive lines as short as 20 nanoseconds. The time required to complete the switching of a single such cell is on the order of 0.1 to 1 microseconds. Devices currently available include arrays of 48 × 48, or 128 × 128 elements. A

256 × 256 device is being evaluated.

Another alternative to realizing the switchable mask array is by means of an array of ferroelectric ceramic optical switches made of the electro-optic material PLZT. These devices are electrically addressed; the applied voltage induces birefringence in the material, which translates into polarization rotation of transmitted light. The switching time of these devices is on the order of a few microseconds. PLZT has been used by NTT in Japan to realize a fiber-optic 4 × 4 crossbar matrix switch.[9] A disadvantage of PLZT is that the material has no memory and high voltages ($>$ 500V) must be used to obtain an adequate electric field for switching.

A final example of SLM technology are acoustic-optic (AO) cells operated in the Bragg diffraction mode.[2,4] These AO devices—among the most highly developed of all SLMs—are used extensively in optical signal processing systems for tasks such as spectrum analysis. AO cells can be used for both modulating the intensity of light and for deflecting a light beam over a range of angles.

Typical AO cells are shown schematically in Figure 2. A single channel electrical input drives a piezoelectric transducer at the bottom of the device, creating an acoustic wave that propagates upward. The device is made of an optically transparent material such as lithium niobate, and the propagating acoustic wave induces a spatially periodic variation of the index of refraction (a moving phase grating), which can be used to modulate or deflect an incident beam. The intensity of the transmitted light can be controlled by increasing the modulation of the induced grating.

The AO device can be also be used as a beam deflector, because the sine of the angle of diffraction is proportional to the temporal frequency of the driving electronic signal or equivalently, the spatial frequency of the moving grating. The left side of Figure 2 shows an AO deflector in which propagating waves of two different spatial frequencies produce deflected beams at two angles, θ_1 and θ_2. Typical electrical signal bandwidth of these devices is one gigahertz, with diffraction angles extending to 10 degrees.

The right side of Figure 2 shows an AO modulator, which produces diffracted beam intensities I_1 and I_2 proportional to the depth of modulation in a constant spatial frequency grating. When used as a modulator, typical AO cells have the spatial resolution necessary to modulate up to 1000 individual beams. When used as a deflector, a cell whose entire aperture is filled by an incident beam can be deflected to 1000 resolvable deflection angles, or a cell illuminated by 30 separate resolvable beams can deflect each beam into 30 resolvable angles.

Finally, if a weighted sum of several sinusoidal electrical signals is input to the AO cell, several diffracted beams will result, each propagating in different directions depending on the input frequency. This mode of operation is useful for a limited broadcasting of a signal in a propagation direction, although it is restricted in practice by nonlinear interactions in the AO cell.[2] Although AO cells are single-channel devices that serve as a spatial delay line to input signals, several independent devices, each with its own input signal line, can be stacked (vertically out of the device plane shown in Figure 2) to achieve multichannel operation.

Recent research has considered various optical devices that offer the possibility of very high speed switching (as low as 10^{-12} seconds) and bistable memory, along with switching energies competitive with electronic systems.[4] Current research in this area concerns developing new materials and structures with reduced switching energies, and constructing large planar arrays of these devices. Much additional work is needed to explore the applications of these devices in optical computing and interconnections.

Optical concepts for interconnection networks

We have studied several techniques for the implementation of generalized crossbar interconnections based on optical matrix-vector or matrix-matrix processors, summarized as follows.

Crossbar interconnections with a matrix-vector or matrix-matrix multiplier. Over the past few years, several different designs for optically implementing linear matrix-vector operations have been proposed. These systems perform operations of the form

$$\underline{c} = \mathbf{A}\,\underline{b} \qquad (1)$$

where $\underline{c}$ is an $N \times 1$ output column vector, $\mathbf{A}$ is an $N \times P$ matrix of weights, and $\underline{b}$ is an input $P \times 1$ column vector. In these systems, the multiply and summation is performed by analog techniques. The numerical values of the input data $\underline{b}$ and the output data $\underline{c}$ must be represented within the system as non-negative intensity values. The individual weights in the matrix $\mathbf{A}$ must be represented by intensity transmittance values, limited to the range [0,1]. Although techniques exist for encoding negative and complex numbers into a non-negative form suitable for processing in optical analog matrix-vector processors, operations involving non-negative real operations on non-negative real data are most easily performed by such systems.

A crossbar is realizable as a matrix-vector multiplier whose processing matrix $\mathbf{A}$ contains weights having only values 0 or 1.[10] For simplicity, assume that $\mathbf{A}$ is square, of dimension $N \times N$. $\mathbf{A}$ is then a generalized permutation matrix whose N^2 entries exactly correspond to the N^2 switches in Figure 1. A value of 1 in row i and column j of $\mathbf{A}$ means that a switch connecting input line j with output line i is closed. Input binary data on N parallel lines is represented by the value of the non-negative signal in each component of the N-vector $\underline{b}$, with a light pulse representing binary 1 and no pulse representing binary 0. If $\mathbf{A}$ has only a single 1 entry in each row and column, it represents a one-to-one permutation of input lines to output lines.

If a particular system can perform matrix operations having two or more 1 entries in each column of $\mathbf{A}$, then broadcasting of one input to several outputs can be performed. If the system permits two or more 1 entries in each row of $\mathbf{A}$, then many-to-one operations such as wire-OR or funneling operations are allowed, provided that the physical output system has hardware and protocols available to resolve simultaneous inputs from two or more sources. As an example of these ideas for $N = 4$, consider

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,1\,0\,0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad (2)$$

which shows the routing of binary bits on four input lines to four output lines. If the optical system can perform matrix-matrix multiplications, then $N \times 1$ output vector $\underline{c}$ generalizes to an $N \times M$ matrix $\mathbf{C}$ and $N \times 1$ input vector $\underline{b}$ generalizes to an $N \times M$ matrix $\mathbf{B}$. In this case, the optical system can perform the interconnection of

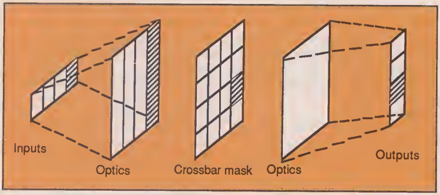


Figure 3. N^2-parallel matrix-vector inner product processor.

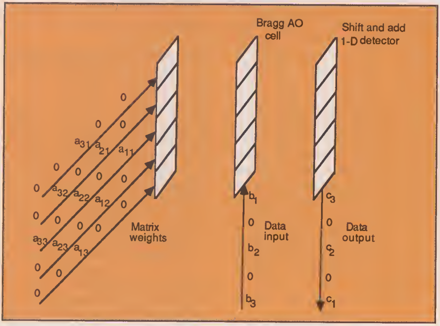

Bragg AO cell
Shift and add 1-D detector
0
0 0
a_{31} a_{21} a_{11}
0 0 0 0
0 a_{32} a_{22} a_{12} 0
0 0 0 0
a_{33} a_{23} a_{13}
0 0
0
Matrix weights
b_1
0
b_2
0
b_3
Data input
c_3
0
c_2
0
c_1
Data output

Figure 4. N-parallel systolic matrix-vector processor.

N parallel input lines, each M bits wide.

Many types of optical matrix-vector and matrix-matrix multipliers may suit crossbar interconnection networks.[11] In some of these systems, the physical switch may be passive in the optical sense (a switchable reflective or transmissive element), providing a pathway for optical signals generated externally. A passive switch avoids costly detection, amplification, and regeneration of optical signals, although it has unavoidable attenuation that may severely limit its performance in some applications.

In a passive network, synchronism between data bits and changes of network switch settings is not required. Moreover, the data bandwidth is limited only by the network attenuation and the optical sources and detectors used. In other systems, the matrix-vector multiplication is implemented by active electro-optical or acousto-optical SLMs. They provide gain, implying the detection and regeneration of optical signals. In such systems, the internal switching mechanism may limit the data bandwidth.

N^2-parallel matrix-vector inner product processor. Figure 3 schematically shows a system capable of performing optical matrix-vector multiplications of the form of Equation (1) in what is called an inner product format.[11] The N input lines drive an array of N light-emitting diodes (LEDs) or diode lasers (labeled "inputs" in Figure 3) with a binary signal, so that a binary 1 is represented by light of a fixed intensity, and a binary 0 is represented by a lower (or zero) intensity. An optical system to the right of the inputs spreads the light from each input source into a vertical column that illuminates the crossbar mask shown.

The crossbar mask consists of an $N \times N$ array of windows representing the entries in the $N \times N$ permutation matrix **A**. An entry of 0 in **A** corresponds to zero light transmission, or an opaque window in the mask. An entry of 1 in **A** corresponds to full light transmission, or an open window. Here, the crossbar mask is an electrically-controllable SLM shutter array that is time-variable through the application of external electrical control signals. Several different technologies mentioned earlier can be used for this purpose, including electro-optic and magneto-optic.

Following the crossbar mask, the next set of optics collects the light transmitted by each row of the mask and transfers the row to the outputs, consisting of a vertical array of N photodetectors corresponding to the N output lines. Thus the system performs a parallel matrix-vector multiplication. Many possible ways to actually implement this schematic design exist, some using discrete optics or optical waveguides.

This N^2-parallel inner product architecture has an optical light efficiency of $1/N$ at most, when used for one-to-one crossbar operation. This occurs because $(N-1)/N$ of the light from each input source does not pass through the crossbar mask. A matrix-vector operation is completed in just one clock cycle. A matrix-matrix multiply for data M bits wide takes M clock cycles. Since it is a passive interconnection, once the mask is set the data can flow through synchronously or asynchronously, can be analog or digital, and has a bandwidth limited only by the sources and detectors. It requires a 1-D array of sources (diode lasers or LEDs) with parallel electronic input lines, and a 1-D array of detectors with parallel electronic output lines; or a method of coupling to input and output fibers. Its reconfiguration time depends entirely on the speed of the individual shutters and the parallelism with which they can be addressed. With existing technology such as the PLZT electro-optic material or the Sight-Mod magneto-optic shutter array, each shutter can open or close in a time on the order of one microsecond.

Another advantage of this system is that no input or output buffers are required; many other systems require them. Broadcasting is possible, allowing a generalized crossbar interconnection. The control is external and could be optical or electronic.

This architecture is useful primarily for circuit switching applications. Efforts at developing this architecture are underway at several companies, including Hughes Research Laboratories and Optivision, Inc.

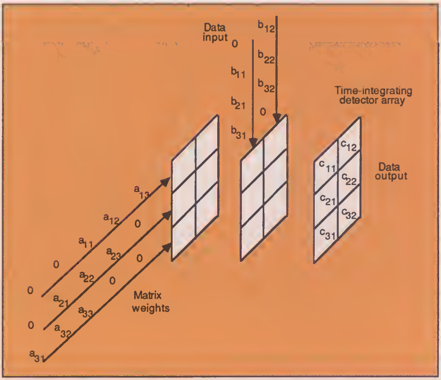


Figure 5. N^2-parallel engagement matrix-vector processor.

Systolic architectures. Systolic architectures have the advantage of providing for rapid reconfiguration of the network, i.e., they can be reconfigured at the bit rate of the data (neglecting any overhead in calculating the needed states of the crossbar matrix elements). In systolic systems, though, the input data must be configured appropriately for the matrix multiplication to proceed. This requires electronics at the input that can buffer the signals and send them into the switch in the appropriate sequence and at appropriate times. In some cases, only minimal buffering is required, such as when the data is originally time-division multiplexed bit by bit.

An N-parallel systolic matrix-vector multiplier, shown schematically in Figure 4 for $N = 3$ and $M = 1$, implements the matrix-vector multiplication of Equation (1).[11] Additional system components such as lenses and apertures are omitted for clarity. The matrix elements a_{ij} conceptually flow in from the left with interleaved 0's as shown. The SLM for the matrix weights can be realized physically by a multichannel AO device or a 1-D array of LED's. The vector elements b_j (also interleaved with 0's) enter the end of another AO cell in time sequence and propagate upward.

The signals are synchronized so that a_{11} is emitted from the center of its SLM when b_1 reaches the center of its AO cell. The system is arranged so that the light from the matrix weight SLM passes through the corresponding pixel of the AO cell and results in the product $a_{ij}b_j$ of these elements at the corresponding element of the 1-D detector array.

The detector is a shift-and-add (such as a 1-D CCD) that accumulates the appropriate elements and outputs the resulting vector elements c_i time-serially. For example, in the first cycle $a_{11}b_1$ will appear at the middle detector. In the second cycle $a_{11}b_1$ will have been (electronically) shifted down one detector, element a_{12} will appear on the corresponding LED, and b_2 will have moved up to the corresponding AO cell location, resulting in $a_{11}b_1 + a_{12}b_2$ on this detector. In this manner, all terms of c_1 are accumulated by the time it reaches the end of the detector array. The entire operation is exactly analogous to electronic systolic matrix-vector multipliers.

Compared to the N^2-parallel inner product architecture, systolic architectures yield crossbars that are more light efficient and have shorter reconfiguration times but lower bandwidth. For an N-parallel systolic system, with sufficiently fast input and output devices, the AO cell will generally limit the bandwidth of each individual line to $\nu_0/2N$, where $\nu_0 \approx 1$ GHz. However, a number of these 1-D elements in the N-parallel systolic multiplier can be placed in parallel, permitting each line to be M bits wide and thereby increasing the effective bandwidth by a factor of M ($M \leq 100$ with current devices). If $M = N$, then the system is an N-parallel systolic matrix-vector multiplier. Because a new matrix is essentially read in for each new bit or word, the system can be reconfigured rapidly.

An N-parallel systolic matrix multiplier requires $N(4N-3)$ clock cycles to complete one matrix-matrix multiplication, or one arbitrary switching operation on N input lines, each N bits wide. In this time one (N-bit) word on each line passes through the network. An N^2-parallel systolic arrangement can do this in $(4N-3)$ clock cycles.

The detector for systolic processors is a shift-and-add detector, instead of the N-parallel readout detector needed in the inner-product multiplier. In most cases, these detectors will limit the data bandwidth.

The input data must be formatted correctly in these systems. Here we distinguish between functional input lines, which have data streams presumably originating from the same (electronic) location, such as the lines in a functional diagram of a network, and physical input lines, which can be located in a physical (optical) setup and carry data that may or may not all be on the same functional line.

In the N^2-parallel case, each successive data input bit in a word must be delayed two units of time and shifted to the next physical input line. In addition, there is a skew from one functional input line to the next of one time unit. In the N-parallel case, all the data is time multiplexed onto one input line, with interleaving 0's. Thus

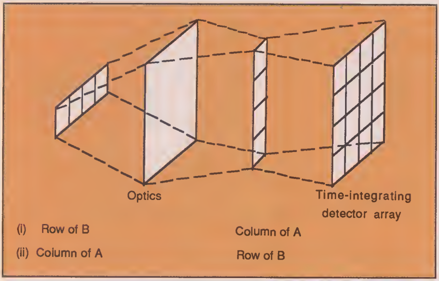


Figure 6. N^2-parallel outer product processor. (i) and (ii) refer to two modes of data entry.

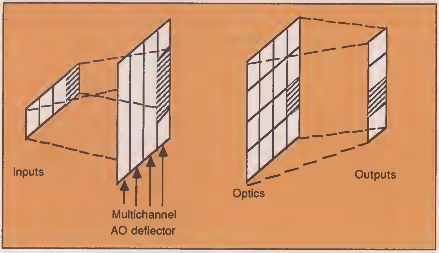


Figure 7. N^2-parallel inner product AO deflector.

one potential application is for switching time-division multiplexed signals. However, this limits the bandwidth to fairly low values (by optics standards), and the bandwidth of each line decreases as N increases.

The operation of systolic systems is strictly synchronous. Also, the light efficiency can be substantially higher than the N^2-parallel inner-product processor discussed above. Systolic systems are valuable when the number of lines is not too large, the bandwidth requirements are not extreme, and the reconfiguration is rapid.

Engagement architectures. Optical engagement architectures use 1-D or 2-D time-integrating (instead of shift-and-add) detectors, preferably with parallel outputs to implement matrix operations.[11] A schematic engagement architecture for matrix multiplication is shown in Figure 5 for $N = 3$ and $M = 2$. The matrix elements of **A** enter into a 2-D source array or a multichannel AO cell, skewed in time and formatted as shown. The data, or elements of **B**, enter a second, crossed multichannel AO cell from the top and propagate downwards. A 2-D time-integrating detector array then accumulates the results and outputs them in parallel.

An N-parallel engagement matrix multiplier requires $N(3N-2)$ clock cycles to complete one matrix-matrix operation or one arbitrary switching operation on N N-bit wide lines, approximately the same as the (N-parallel) systolic case. An N^2-parallel engagement arrangement takes $(3N-2)$ clock cycles.

The input data formatting requirements resemble the systolic case, but without interleaved 0's. We can think of the width of each functional input line as equal to the word length, so that an M-bit wide line can transfer one word at an instant of time. The N^2-parallel engagement system has N physical input lines, and the switch operates on word slices: it is word serial and bit parallel except for a unit time shift from one physical line to the next. In other words, the first line has the first bit of each word, sequentially in time, and so forth. In the systolic processor described previously, the physical lines are actually serial over diagonals of matrix **B** rather than strictly bit serial or word serial. As with systolic systems, reconfiguration is rapid but updating the state of the switches is needed even when the state does not change from one multiply to the next.

N^2-parallel outer product processor. An N^2-parallel outer product processor such as that shown in Figure 6 performs the matrix-matrix operation $\mathbf{C} = \mathbf{AB}$ by taking outer products of column i of **A** and row i of **B**, and summing the results over i. It can perform a matrix-matrix operation in N clock cycles. In some implementations the data **B** and the switch matrix **A** can both be read in rapidly, offering a potentially high bandwidth.

There are two variations of the system. In one variation, a row of **B** is input at the left through an optical source array, or through a horizontally oriented AO cell illuminated by an external light source. At the same time, a column of **A** is entered by a second linear array SLM; another AO cell could be used for this purpose. A set of optics forms the outer product between the two input vectors, and the 2-D time-integrating detector array accumulates the output matrix **C**. The light efficiency of this system is $\leq 1/N$.

In a second variation of the system, the SLM at the left inputs a column of **A** while the second SLM inputs a row of **B**; its light efficiency is high. For large N, the 2-D detector array is the bandwidth-limiting component.

Broadcasting with this system is easy, providing a generalized crossbar. As in the systolic and engagement cases, the data is

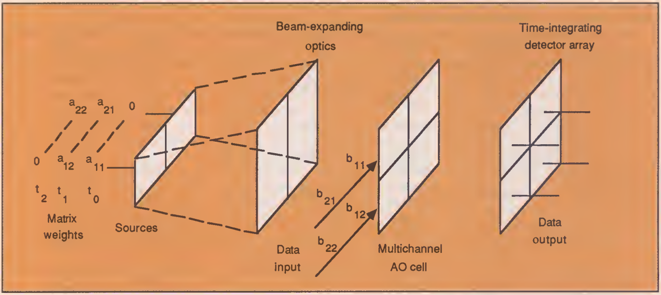


Figure 8. N^2**-parallel inner product/engagement processor.**

completely synchronous. Since the **A** matrix is read in for each matrix multiplication, reconfiguration is rapid, although the state of the switches needs to be updated for each matrix multiply.

N^2**-parallel inner product AO deflector.** While the above systems may be useful in many applications, they all share a common fault. Since the matrix **A** is mostly 0's (in fact it has a minimum of $N(N-1)$ 0's and at most N 1's for an $N \times N$ matrix when used for one-to-one interconnections), most of the scalar multiplications performed by the above systems are multiplies by 0. This translates into either wasted time (thereby lowering the bandwidth) or wasted light (thereby lowering light efficiency). The N^2-parallel inner product AO deflector shown in Figure 7 resembles the N^2-parallel inner product system and alleviates some of its problems.

The input data to the crossbar enters through the 1-D array of sources on the left. Each source then illuminates a corresponding cell (or channel) of a multichannel AO device used to deflect the light vertically. A separate channel exists for each data-input line, or each column of **A**. Rather than reading the elements of **A** into the AO device (N elements into each channel), one single number is input into each channel and the signal is frequency-modulated according to the amount of vertical deflection desired for the corresponding input line.

Broadcasting can be achieved, to a limited extent, by superimposing multiple frequencies onto the same channel of the AO cell, or by time multiplexing the different frequencies in the cell. Each channel then deflects the signal the appropriate amount in the y dimension, and the light is focused horizontally in the x dimension, yielding a one-dimensional output array. Thus, the second half of the optical system resembles that used in the N^2-parallel inner product system.

This system is passive, thus its bandwidth is generally limited by external sources and detectors, not by the multichannel AO cell. A bandwidth of one gigahertz should be achievable. The outputs are detected by a real-time 1-D array. The reconfiguration time of the network equals the time window used in each cell, which is less than or equal to the maximum time window or fill time of the device. Thus we can expect a reconfiguration time the order of one microsecond.

Another advantage of this system is that, as in the N^2-parallel inner product multiplier, there are no special formatting or buffering requirements on the input or output data. The data enters in a word-parallel, bit-serial fashion; each different functional input line is a different physical input line. The N control signals are input in parallel. However, electronic hardware is required to generate the frequency-modulated control signals. The number of input and output lines equals the number of cells in the AO device, which limits the number of lines that can be switched with one crossbar. The light efficiency should exceed $1/N$ and is limited by the characteristics of the AO cell and the imaging optics that follow it. The operation of the switch can be synchronous or asynchronous, and the data can be analog or digital.

N^2**-parallel inner product/engagement processor.** Another type of matrix-matrix processor architecture that could be used as an optical crossbar is shown in Figure 8. The system is a combination of an N^2-parallel inner product processor and an N^2-parallel engagement architecture. The input data (b's) enter a parallel array of AO cells as shown, although the data for each row enters simultaneously instead of being staggered as in the N^2-parallel engagement processor. The matrix weights also arrive in a time-staggered form and control the illumination of LED's or diode lasers as shown. A set of optics similar to that in the N^2-parallel inner product processor spreads the light across the multichannel Bragg cell, and the final result accumulates on a time-integrating detector array. The results emerge in a time-offset fashion from a row of the detector array.

This system requires $(3N-2)$ clock cycles to complete a matrix-matrix multiplication and requires a synchronous data format. Broadcast operation is possible, and the overall light efficiency can be higher than $1/N$. This is an active inter-

Table 1. Comparison of optical crossbar capabilities.

Method of Implementation	Number of Lines (N)	Bandwidth (BW)	Reconfiguration Time	Broadcast	Data Format	Type of Detector
N^2-parallel Inner Product	>256	100 MHz - 1 GHz (passive)	(N)1 μs	Yes	Async	Real time $N \times 1$ element
N^2-parallel Systolic	>100	10 MHz*	N/BW	Yes	Sync	$N \times N$ shift/add
N^2-parallel Engagement	>100	10 MHz*- 100 MHz	N/BW	Yes	Sync	$N \times N$ time-integrating
N^2-parallel Outer Product	>256	10 MHz*- 100 MHz	N/BW	Yes	Sync	$N \times N$ time integrating
N^2-parallel Inner Product AO Deflector	>100	100 MHz- 1 GHz (passive)	1 μs	Limited	Async	Real time $N \times 1$ element
N^2-parallel Inner Product/ Engagement	>100	10 MHz*- 100 MHz	N/BW	Yes	Sync	$N \times N$ time integrating

*Limited by output (detector array) electronics. Higher numbers (when given) apply when fibers guide light to discrete detectors or other optical components. BW is the signal bandwidth per line.

connection network, and for large N the bandwidth of each line is limited by the detector electronics. The reconfiguration time is similar to that of systolic and engagement processors.

Performance comparison

The optical systems introduced in the previous section differ widely among themselves in many important characteristics when used for interconnection. For example, some of the systems provide a bandwidth essentially independent of the size of the system, while for others the bandwidth is an inverse function of the system size. In this section, we compare the different systems based on six parameters of interest in crossbar implementation. The results are summarized in Table 1.

(1) *Number of lines.* We feel that the practical limits of optics, SLM, and detector technology allow the implementation of moderately large crossbars (at least 100 × 100).

(2) *Bandwidth.* For passive optical networks, the bandwidth of each line is limited by optical transmitters, receivers, and the attenuation of the interconnection network. Existing individual sources (laser diodes) can operate at 10 gigahertz or more, and detectors can operate at greater than one gigahertz. Arrays will generally be slower due to power dissipation and optical and electrical crosstalk. These components make up optical transmitters and receivers that must encode the binary information and transmit it reliably.

With a fixed transmitter/receiver pair, the bit error probability will degrade as the passive attenuation increases. Although a bit error probability of 10^{-9} is assumed for telecommunications systems, interprocessor communication may require effective bit error probabilities of 10^{-12} or less. This performance can be achieved by improved transmitters or receivers, by lower passive attenuation in the optical interconnection network, or by error-correcting coding.

Error-correcting coding effectively reduces the data bandwidth available. For the entries in Table 1, this bandwidth is defined for each line as distinct from the total bandwidth of the entire interconnection system. Electronic systems typically operate at 10-100 megabits per second for each line. Active optical networks are mainly limited by the SLMs used for modulation and beam deflection; in many systems the electronic bottlenecks in the detector array cause a more stringent limit.

(3) *Reconfiguration time.* Reconfiguration of optical networks is limited to approximately one microsecond for moderate or large networks with current or near-future technology. In fact, most 2-D switch arrays have a frame time of one millisecond or greater. Most 2-D optical device arrays presently have switching times of one microsecond for each switch at best. Acousto-optic devices have faster response times, but the electronic input (to each cell) is serial and limited to a few gigahertz, and the signal cannot be changed as it propagates down the cell.

Most of the acousto-optic architectures permit fast reconfiguration at the expense of bandwidth. For an electronically controlled optical switching array, it is a matter of switching time *and* addressing schemes to control all of the switch states. For an array of N^2 switches, N^2 lines are needed to change all switch states in parallel. This is impractical for large arrays. Optical control can provide parallel control, or the switches could be addressed by N lines coming in one side and N lines coming in the other. In the latter case a signal on row i and a signal on column j could address switch (i,j). This would require N steps to reconfigure the entire array.

We should mention that devices (SLMs) with both optical control and electronic control as described above do exist; frame times measure approximately 100 microseconds to 100 milliseconds. Electronic switches can typically switch in 50 to 100 nanoseconds (for each switch).

The total reconfiguration time of a network depends on addressing schemes for circuit switched networks and on decoding and switching for packet switched networks. Fast reconfiguration of the network is needed in an environment where the switching permutations change dynamically. However, slow switching could be tolerated in applications where each connection is followed by the transfer of large blocks of data.

(4) *Broadcast capability.* If the interconnection network can achieve broadcast in one step, we can reduce the overall time-complexity of many parallel algorithms. In electronic implementations, the provision for broadcasting involves higher control complexity, a larger number of pins, and/or slower operation. In some optical systems, the addition of broadcast capability involves only minor increases in complexity.

(5) *Data format.* The switching process as well as the transfer of data can be performed either synchronously or asynchronously. The synchronous operation of active interconnection networks requires strobing or capture of data at precise instants of time in the system; a master clock is required to control system timing. Most passive systems are asynchronous systems and do not suffer from this limitation.

(6) *Type of detector.* Implementation of these systems requires various kinds of multichannel optical detectors. The inner-product processors need a vector array of detectors with separate channels and a real-time parallel output.

The data bandwidths of these systems are generally limited by the response characteristics of the detector array. In some systems the output may be directly coupled to a light guide or fiber. In this case the detectors, if any, lie some distance from the optical switching array. Because the limitation in the bandwidth of detector arrays results primarily from electrical crosstalk, this coupling can permit substantially higher data bandwidths than could be achieved using detector arrays. For example, routing the light to discrete detectors permits higher bandwidth at the expense of increased size and hardware.

The engagement and outer-product systems with full N^2 parallelism require $N \times N$ time-integrating detectors that can operate synchronously with the data. The 2-D $N \times N$ detector arrays have either N output lines or one output line. Thus the necessity of multiplexing at least N detector signals onto each electronic output line creates a bottleneck and limits the bandwidth of the overall system. Using N^2 parallel outputs could eliminate this bottleneck; the feasibility of such detectors deserves investigation. Also, the number of channels N will be limited by the time-integration of noise.

The systolic architectures require a shift/add detector array to perform the summation. $N \times N$ shift/add arrays necessarily have at most N output lines. This and the use of electronics to perform the shift and add limits the detector speed, in turn lowering the overall bandwidth of the system.

Once again, Table 1 summarizes the above considerations for the six basic optical matrix-vector crossbar architectures described in this article.

Other factors to consider are the control mechanisms for the switches (optical or electronic); the optical efficiency of the switches and its effect on error rate and transmission signal-to-noise ratio; the propagation delay time of each switch and effects on the overall network delay; the volatile or nonvolatile nature of various optical switch technologies; and overall network reliability as a function of topology and the particular devices and components used.

Passive optical interconnection networks must be extremely efficient in using the available photons, particularly if they have broadcast capability. In the N^2-parallel system, the input signals fan out into N beams, thus the system has an insertion loss of at least 20 decibels per channel for $N = 100$. We must carefully design practical systems so that optical fan in[12] following the switching SLM and other losses do not degrade performance.

The main observation here is that with existing applicable technology optical switching is slower than electronics, although the signal bandwidth of optics is much greater. This result, along with the inherent parallel characteristics of optics, leads to the conclusion that optical interconnection systems are most useful when the reconfiguration time greatly exceeds the time duration of data packets to be transferred, and when there are sufficient processors and parallelism for many simultaneous data transfers. These conditions are true for block-oriented data and large file transfers, such as in graphics and image processing.

We have described several possible bulk optical systems for implementation of crossbar networks and studied the tradeoffs involved. Some of these systems show promise for use in large multiprocessor systems. Advantages of these systems include a large amount of inherent parallelism, high data bandwidth, small size and power requirements, and relative freedom from mutual interference of signals. The main difficulties we need to surmount are the slow reconfiguration, means for efficient conversion of electronic signals to optical signals and vice versa in large arrays, and techniques for control.

We could tolerate slow switching in applications where each connection is followed by the transfer of large blocks of data, as in image processing. Current electronic interconnection networks can change the state of a single switch rapidly (in 50 to 100 nanoseconds), although the bandwidth of each line is low (10 to 100 megabits per second).

Optical interconnection networks at present generally have complementary capabilities: very large bandwidths per line (greater than one gigabit per second) and slower individual switching times (one microsecond per switch). A significant amount of work is still required to bring these systems to the stage of commercial availability.

We have found that moderately large crossbars (greater than 100×100) may be feasible using near-future optical technology. An intriguing and powerful possibility involves using these crossbars as building blocks to make larger crossbar networks. To do so, we need to devise a means of cascading these optical crossbars.[7] This involves studying the input

Late Magazines?
No Magazines?
Membership Status Problems?
No Answers To Your Complaints?

Let your Computer Society Ombudsman cut through the red tape for you.

Computer Society
Ombudsman
IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

and output formats of the data in the different optical systems for compatibility. We must also consider the light efficiency of the crossbar, as it determines how often detection and regeneration of the optical signals are needed. We must devise methods of physically coupling the output of one system to the input of the next. Additionally, the signals may need to be sent to multiple crossbars in different locations. Of course, one way of doing this would be to use electrical lines to interconnect the optical crossbars, but it is preferable to keep as much of the overall system optical as possible. This maximizes the advantages that the optics provide, such as high bandwidth and large number of lines.

All-optical amplifiers have been demonstrated.[4] While practical problems remain in building 1-D arrays of optical amplifiers, it can, in principle, be done.[3] Optical cascading of crossbars can be done with fibers or holograms. It has been shown[3] that holograms can be quite powerful for interconnecting optical gates. Even the capability of connecting a small number (four or five) of sizable crossbars together could yield a system substantially more powerful than a single crossbar. We are currently studying these and other problems in the implementation of optical crossbar networks. ☐

Acknowledgment

This research is supported by DARPA/ARO Contract No. DAAG29-84-K-0066.

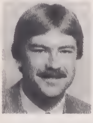
References

1. M.J. Flynn, "Very High-Speed Computing Systems," *Proc. IEEE*, Dec. 1966, pp. 1901-1909.
2. Special issue on optical computing, *Proc. IEEE*, July 1984.
3. A.A. Sawchuk and T.C. Strand, "Digital Optical Computing," *Proc. IEEE*, July 1984, pp. 758-779.
4. T.E. Bell, "Optical Computing: A Field in Flux," *IEEE Spectrum*, Aug. 1986, pp. 34-57.
5. T. Burggraff et al., "The IBM Los Gatos Logic Simulation Machine Hardware," *Proc. Int'l Conf. Computer Design*, 1983, pp. 584-587.
6. M.M. Denneau, "The Yorktown Simulation Engine," *Proc. 19th Design Automation Conf.*, Las Vegas, 1982, pp. 55-59.
7. G. Broomell and J.R. Heath, "An Integrated-Circuit Crossbar Switching System," *Proc. 4th Int'l Conf. Distributed Computing Systems*, May 1984, pp. 278-287.
8. A.R. Tanguay, Jr., "Materials Requirements for Optical Processing and Computer Devices," *Optical Engineering*, Jan./Feb. 1985, pp. 2-18.
9. A. Himeno and M. Kobayashi, "4 × 4 Optical-Gate Matrix Switch," *J. Lightwave Tech.*, Vol. LT-3, April 1985, pp. 230-235.
10. B. Clymer and S.A. Collins, Jr., "Optical Computer Switching Network," *Optical Engineering*, Jan./Feb. 1985, pp. 74-81.
11. R.A. Athale, "Optical Matrix Algebraic Processors: A Survey," *Proc. 10th Int'l Optical Computing Conf.*, April 1983, pp. 24-31. IEEE catalog no. 83CH1880-4.
12. J.W. Goodman, "Fan-In and Fan-Out with Optical Interconnects," *Optica Acta*, Dec. 1985, pp. 1489-1496.

Alexander A. Sawchuk is director of the Signal and Image Processing Institute and a professor in the Department of Electrical Engineering at the University of Southern California. His professional interests include optical information processing, machine vision, and image analysis.

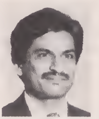
Sawchuk is a fellow and chair of the Technical Council of the Optical Society of America, and a fellow of the Society of Photo-Optical Instrumentation Engineers. He is also a senior member of the IEEE.

Sawchuk received the BS degree in electrical engineering from the Massachusetts Institute of Technology in 1966 and the MS and PhD degrees from Stanford University in 1968 and 1972, respectively.



B. Keith Jenkins is a research assistant professor at the University of Southern California. His research interests include optical digital numeric and symbolic computing, optical interconnections, parallel computation models and complexity, computer-generated holography, and optical 3-D position-sensing systems.

Jenkins received the BS degree in applied physics from the California Institute of Technology in 1977 and the MS and PhD degrees in electrical engineering from the University of Southern California in 1979 and 1984, respectively.



C.S. Raghavendra is an associate professor at the Department of Electrical Engineering at the University of Southern California. His research interests include computer system architecture, fault-tolerant computing, optical computing, and parallel and distributed systems.

Raghavendra received the Presidential Young Investigator Award in 1985.

He received a BS degree in physics from Bangalore University in 1973, BE and ME degrees in electronics and communication from the Indian Institute of Science, Bangalore, in 1976 and 1978, respectively, and a PhD in computer science from the University of California, Los Angeles, in 1982.

Anujan Varma is a member of the research staff in the system interconnection structures department at the IBM Thomas J. Watson Research Center. His research interests include computer architecture, fault-tolerant computing, interconnection networks, and optical computing.

Varma received a BS degree in electrical engineering from the University of Calicut and an ME degree in automation from the Indian Institute of Science, Bangalore. He holds a PhD degree in computer engineering from the University of Southern California.

Readers may write to Sawchuk at the Dept. of Electrical Engineering—Systems, MC 0272, University of Southern California, Los Angeles, CA 90089.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC

IEEE

THE COMPUTER SOCIETY
OF THE IEEE

ADVANCE PROGRAM

SYMPOSIUM ON THE SIMULATION OF COMPUTER NETWORKS

August 4-7, 1987 the Antlers Hotel Colorado Springs, CO

SPONSORS

IEEE/CS Technical Committee on Simulation, Dr. Sallie Sheppard, Texas A&M, Chair
ACM Special Interest Group on Simulation, Mr. Paul Roth, Virginia Polytechnic, Chair

TECHNICAL PROGRAM

Technical sessions will be held Wednesday, Thursday, and Friday, August 5, 6, and 7, 1987. There will be 18 to 27 papers in the technical program arranged in nine sessions listed below. There may be one or two panel sessions. We do not expect to go to parallel sessions.

Standardized Networks
Satellite Comm Networks
Local Area Network Simulation
Simulation Test Beds
Computer Bus Architectures
Performance Evaluation
Network Architectures
Network Partitioning
Distributed Network Design

COMMITTEE

Dr. Ashok Agrawala, U of Maryland
Dr. William Hawe, Digital Equipment Corporation
Mr. Carl Manson, Architecture Technology Corp
Dr. S. Ron Oliver, SAIC
Dr. Roy Wnek, Booz, Allen, Hamilton
Dr. Jim Browne, U of Texas
Dr. Edward Lazowska, U of Washington
Mr. Dale Murray, General Instruments
Dr. John Sanguinetti, Dana Computer Corp
Dr. Jacob J. Wolf III, Colorado State U

REGISTRATION

All attendees are expected to bear all costs including the registration fee. This is $175 for ACM and IEEE members before 6 July 1987, $225 thereafter. Rates are $50 higher across the board for non-members. Registration includes proceedings and coffee breaks. Registration will begin on Tuesday evening, August 4, at the hotel.

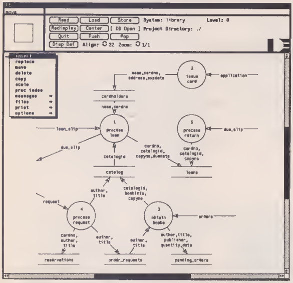
For registration information, contact Dr. Charles Shub, Local Arrangements Chairman, U of Colorado, Computer Science Dept, Colorado Springs, CO 80933 (303-593-3325).

HOUSING

A block of rooms has been reserved at The Antlers, Chase Stone Center, Colorado Springs, CO 80903 (303-473-5600). The conference rate is $65 per person (single or double room).

CHAIRMAN	INDUSTRY CO-CHAIR	UNIVERSITY CO-CHAIR	TREASURER
Mr. Mitchell Spiegel GTE/GSC	Mr. Michael Green Network Systems	Dr. Edgar Sibley George Mason University	Dr. Robert Kinicki Worcester Polytechnic

IDE is the state-of-the-art in computer-aided software engineering



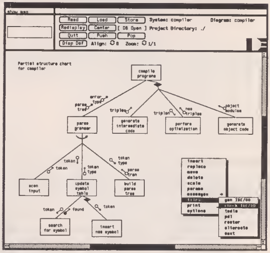
Interactive Development Environments offers a powerful family of graphical editors and an integrated software development environment.

IDE's workstation-based tool environment includes:

- support for several widely used analysis and design methods, including Structured Systems Analysis, Structured Design, and User Software Engineering, with *extensive checking* built into the tools
- the IDE Data Dictionary system, which can be edited directly from the graphical editors; it supports definition of names, along with types, constraints, aliases, and associated text
- rapid prototyping and application generation with the RAPID/USE system
- support for real-time extensions to structured analysis

Software through Pictures™
graphical editors

- Dataflow Diagram Editor
- Structure Chart Editor
- Data Structure Editor
- Entity Relationship Editor
- Transition Diagram Editor



- Visible Connections™, an open software architecture to support extensions to the tools
- a compact and efficient relational database management system to support the project database and IDE Data Dictionary
- consistent user interfaces to the graphical tools
- a powerful tools library to support additional tool and application development
- available now for the Sun, Apollo, and VAXstation Workstations

Interactive Development Environments is pushing back the frontiers of technology in computer-aided software engineering. Our tools represent an *outstanding* value for their performance and capabilities, with substantial discounts available to universities for teaching and research. Contact IDE and see how we can help to improve your software engineering environment. Call us, telex us, or send us electronic mail at ucbvax!sun!ide!sales.

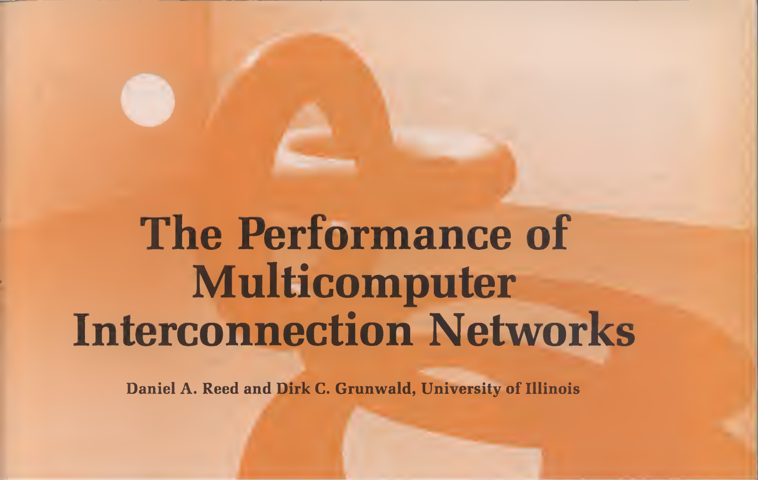
INTERACTIVE
DEVELOPMENT
ENVIRONMENTS

Sales and Information:
Tel: (415) 543-0900

150 Fourth Street, Suite 210 • San Francisco, CA 94103 • Telex: 184 324 IDE

™Software through Pictures and Visible Connections™ are trademarks of Interactive Development Environments, Inc.
VAX station is a trademark of Digital Equipment Corporation.

Reader Service Number 3



The Performance of Multicomputer Interconnection Networks

Daniel A. Reed and Dirk C. Grunwald, University of Illinois

Sculpture Simulation © Paul Brown/Digital Pictures 1983

You cannot find an interconnection network optimal for *all* applications but, by employing the methods described here, you can choose one that fits the requirements of certain classes of applications well.

> The most constant difficulty in contriving the engine has arisen from the desire to reduce the time in which the calculations were executed to the shortest which is possible.
>
> —Charles Babbage (1837)

The spectrum of parallel processor designs can be divided into three sections according to the number and complexity of the processors. At one end there are simple, bit-serial processors. Any one of these processors is of little value, but when it is coupled with many others, the aggregate computing power can be large. This approach to parallel processing can be likened to a colony of termites devouring a log. The most notable examples of this approach are the NASA/Goodyear Massively Parallel Processor,[1] which has 16K one-bit processors, and the Thinking Machines Connection Machine,[2] which has 64K one-bit processors.

At the other end of the spectrum, a small number of processors, each built using the fastest available technology and the most sophisticated architecture, are combined. An example of this approach is the Cray X-MP. This type of parallel processing is akin to four woodmen attacking the log with chainsaws.

The middle approach is analogous to a small army of hungry beavers. It combines a large number of microprocessors. Within the last ten years, several groups have advocated such an approach to parallel processing, one based on large networks of interconnected microcomputers.[3-7] Each node in these multicomputer networks contains a processor with some locally addressable memory, a communication controller capable of routing messages without delaying the processor, and a small number of connections to other nodes.

The applications suggested for multicomputer networks are diverse—they include finite-element problem analysis, partial differential equation solving, linear algebra, physics problem solving, game tree searches, and functional programming. The cooperating tasks of a parallel algorithm for solving one of these problems will execute asynchronously on different nodes and communicate via message passing.

Because its nodes do not share any memory, the multicomputer interconnection network must efficiently support message passing. Typically, multicomputer networks provide datagram message transmission rather than virtual circuits.

0018-9162/87/0600-0063$01.00©1987IEEE

Because of the high latency for message transmission, compared to a memory access, optimization of message traffic patterns is crucial. Methods for evaluating the performance of network alternatives are our subject here.

Commercial multicomputer networks. Recent advances in VLSI have finally made large multicomputer networks both technically and economically feasible. For example, a collaborative project at the California Institute of Technology produced the Cosmic Cube, a multicomputer network based on a hypercube interconnection. A hypercube is a generalization of a cube to D dimensions, where each of the 2^D cube vertices is a multicomputer node. If the hypercube nodes are numbered, beginning with zero, then node i is connected to the D other nodes whose numbers are obtained by negating one bit of the binary representation of i. Figure 1 illustrates the three-dimensional case.

In 1985 Intel announced the iPSC,[8] a commercial version of the Cosmic Cube containing up to 128 nodes. Each iPSC node contains 512K bytes of memory, an Intel 80286 microprocessor, an Intel 80287 floating-point coprocessor, and seven Ethernet transceiver chips to manage internode contains 512K bytes of memory, an Intel 80286 microprocessor, an Intel 80287 similar design.

Both the Intel iPSC and Ametek System/14 require many chips per node and one board for each node, limiting the maximum feasible system size. In contrast, each node of the NCube/ten hypercube[9] contains only seven chips: a custom 32-bit microprocessor with both on-chip floating-point and hardware communication support and six memory chips. Sixty-four nodes occupy a single board, and a fully configured system contains 1024 nodes.

Although the hypercube interconnection matches the communication patterns of many common algorithms, its primary disadvantage is that it requires a logarithmic increase in node connectivity as the number of nodes increases. Moreover, the limited interconnection afforded by VLSI node implementations suggests that other interconnection networks may be more appropriate for large numbers of nodes. Hence, we consider techniques for evaluating network performance.

Overview. Several different characterizations of network performance have been suggested.[4,6,7] However, repeated studies have shown that a system's performance is maximized when all the components are balanced (i.e., when there is no single system bottleneck).[10] Multicomputers are no different; optimizing performance requires a judicious combination of node computation speed and message transmission latency. The rate at which a network routes messages and the rate at which nodes perform computations are intimately related.

Here we examine the interdependency of nodes and multicomputer interconnection networks using simple "back of the envelope" calculations based on the asymptotic properties of queueing networks. Although the interaction of communication and computation can be examined analytically, time-varying behavior and the idiosyncrasies of system software can only be captured by observation and measurement. Thus, we also show how analytic models can be extended to benchmark existing interconnection networks such as the hypercube connection of the Intel iPSC.[8] Before beginning discussion of analytic models, however, we must first define several important terms and concepts used to quantify the performance of networks.

Definitions and background

The maximum internode distance, often referred to as the *diameter* of the interconnection network, places a lower bound on the delay required to propagate information throughout the network. It is simply the maximum number of communication links that must be traversed to transmit a message to any node along a shortest path.

In contrast, the mean internode distance is the expected number of link traversals a "typical" message needs to reach its destination. The mean internode distance is a better indicator of average message delay than the network diameter, but as we shall see, it too fails to completely capture the relative communication capacity of different interconnection networks.

Unlike the network diameter, the mean internode distance depends on the message routing distribution. This routing distribution specifies the probability that different network nodes exchange messages, and it ultimately depends on the communication requirements of the application and system programs as well as on the mapping of these programs onto the network.

In its most general form, the mean internode distance (the number of link visits, or LV) is given by

$$LV = \sum_{l=1}^{lmax} l \cdot \Phi(l) \qquad (1)$$

where $\Phi(l)$ is the probability of an arbitrary message crossing l communication links (i.e., the routing distribution), and $lmax$ is the network diameter. Different choices for $\Phi(l)$ lead to different message routing distributions and, in turn, different mean internode distances. In the following, we consider three different message routing distributions for which it is possible to obtain closed forms for the mean internode distance. To do this, however, we must first distinguish between two types of interconnection networks: symmetric and asymmetric.

In a symmetric interconnection network there exists an isomorphism that maps any node of the network graph onto any other node. Thus, all nodes possess the same view of the network. A bidirectional ring network is a simple example of a symmetric interconnection network because two nodes are always reachable by crossing any given number of communication links, and a simple node renumbering suffices to map any node onto any other. An asymmetric interconnection network is any network that is not symmetric (e.g., a tree). Here, we consider only symmetric networks; for an analysis of asymmetric networks, see Reed and Fujimoto.[11]

Uniform message routing. A message routing distribution is said to be uniform if the probability of node i sending a message to node j is the same for all i and j, $i \neq j$. Because we are interested in message transfers that use the network, we exclude the case of nodes sending messages to themselves.

Uniform routing distribution is appealing because it makes no assumptions about the type of computation generating the messages; this is also its greatest liability. However, because most computations should exhibit some measure of communication locality, it provides what is likely to be an upper bound on the mean internode message distance.

Sphere of locality. Suppose the uniform message routing assumption were relaxed. We would expect any reasonable mapping of a distributed computation onto a multicomputer network to place those tasks

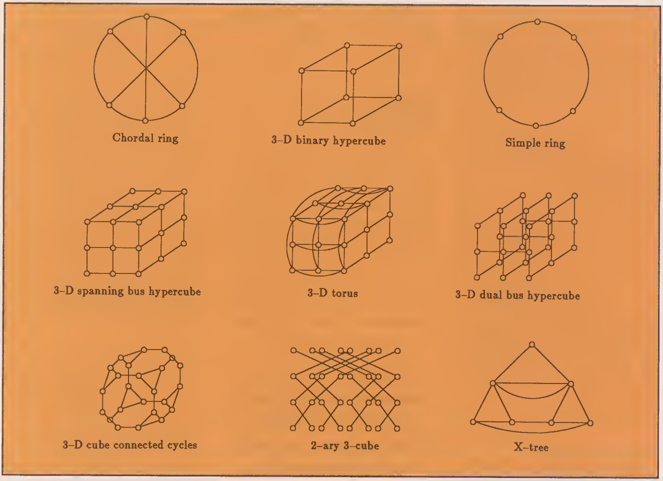


Figure 1. Representative interconnection networks.

that exchange messages with high frequency in close physical proximity. One abstraction of this idea places each node at the center of a sphere of locality with radius L, measured in hops. A node sends messages to the other nodes inside its sphere of locality with some (usually high) probability ϕ, and to nodes outside the sphere with probability $1 - \phi$. This model reflects the communication locality typical of many programs (e.g., the nearest-neighbor communication typical of iterative partial differential equation solvers coupled with global communication for convergence checking[9]).

Decreasing probability message routing. The previous definition of locality is useful if the probability of visiting the locality is high and the size of the locality is small compared to the size of the network. There are, however, many cases in which this is not an appropriate abstraction. An alternate, intuitively appealing notion of locality is that the probability of sending a message to a node decreases as the distance of the destination node from the source node increases. The distribution function $\Phi(l) = Decay\ (d,\ lmax) \cdot d^l$ $0 < d < 1$, where *lmax* is the network diameter and d is a locality parameter, is particularly attractive. Here, *Decay* (*d*, *lmax*) is a normalizing constant for the probability Φ, chosen such that the probabilities sum to one. As d approaches one, the cumulative distribution function of Φ approximates a linearly increasing function of the distance from the source node. Conversely, as d approaches zero, the cumulative distribution function of Φ approaches a nearest-neighbor communication pattern. Small values of d mean that messages travel only small distances (i.e., that locality exists), whereas larger d values mean that messages can travel larger distances.

These three message routing distributions—uniform, sphere of locality, and decreasing probability—can be used to calculate mean internode distances. However, the mean internode distance does not reflect link utilization. For this, we must define visit ratios.

Visit ratios. Each time a node sends a message to another node, the message must cross some communication links and pass through intermediate nodes before reaching its destination node. At the destination, it causes some computation to take place. This datagram model of communication accurately reflects a variety of multicomputer applications, including solution of partial differential equations and functional programming.

Each link crossing and destination node computation constitutes a *visit* to that link or processing element. If all possible source/destination pairs and the probabilities that they exchange messages are considered, the number of visits to each communication link and node by an average message can be calculated. Now consider such an average message and an arbitrary device i (either a node or a link). The average message will visit device i a certain number of times. This mean number of visits is called the visit ratio of device i and is denoted by V_i.[10]

By analogy, imagine that each message deposits a "token" at a device during each visit. After observing the system of nodes and links for some "reasonably long" interval, we can obtain the visit ratios by normalizing the token counts at each device (i.e., by dividing the number of tokens at that device by the total number of token-depositing messages that were created during the observation period). As we shall see, visit ratios can be used to locate those devices in a network that most limit performance—the bottlenecks.

Under the uniform message routing distribution discussed earlier, the visit ratios for all network nodes must be the same. In other words, the probability of visiting for each destination node is the same, and V_{PE}, the visit ratio for the nodes, is

$$V_{PE} = \frac{1}{K} \qquad (2)$$

when there are K nodes.

Somewhat surprisingly, the node visit ratios are also given by Equation 2 when the two proposed nonuniform message routing distributions are considered. This follows from two features of the networks and the routing distributions: network symmetry and similar message routing behavior at all nodes. That is, if all nodes behave similarly and the network is symmetric, each node is equally likely to be visited.

As we saw earlier, the mean internode distance LV represents the average number of visits to all communication links by a message. Dividing this number by the number of communication links yields the communication link visit ratios

$$V_{CL} = \frac{LV}{Numlinks\,(K, Net\text{—}type)} \qquad (3)$$

This quantity can be viewed as a measure of the message intensity supported by a single link. If V_{CL} is near one, then nearly all messages must cross each link at some point along the paths to their respective destinations. Unfortunately, this simple definition is accurate only if the interconnection network contains just one type of communication link (e.g., a ring). In a binary tree one would expect the communication traffic on the links at each tree level to be different, leading to different link visit ratios for each level.

Proposed interconnection networks

A plethora of interconnection network proposals have appeared in the research literature, and an enormous amount of research has centered on the design and analysis of these networks. Among the proposed interconnection networks are several that serve as useful points of reference or have particularly attractive features: the single bus, the complete connection, the simple ring, the chordal ring,[3] the spanning bus hypercube,[7] the dual-bus hypercube,[7] the torus,[7] generalized hypercubes,[4] the cube-connected cycles,[12] the R-ary N-cube,[13] the lens,[6] the X-tree,[5] and the B-ary tree. Figure I illustrates a subset of these networks.

A single bus joins all nodes and uses a contention resolution protocol to resolve simultaneous requests

A single bus joins all nodes and uses a contention resolution protocol to resolve simultaneous requests for the bus. Although inexpensive, it can efficiently support only a modest number of nodes. In contrast, the complete connection directly connects each node to all other nodes. Its cost rises quadratically with the number of nodes, although its performance is the best achievable. These two networks, the bus and the complete connection, bound the spectrum of price and performance for all practical multicomputer networks.

The single ring is a modest improvement over a bus, but it too can efficiently support only a small number of nodes. The chordal ring introduces cross or chordal link connections between nodes on opposite sides of the simple ring, reducing the maximum number of links that must be traversed to reach a destination node.

The spanning bus hypercube is a D-dimensional lattice of width w in each dimension. Each node is connected to D buses, one in each of the orthogonal dimensions; w nodes share a bus in each dimension.

The dual-bus hypercube was proposed to bound the number of connections to each node. It is obtained by pruning $D-2$ bus connections from each network node. In particular, one dimension, the 0th dimension, is distinguished, and all nodes are connected to a 0th-dimension bus. In each $D-1$ hyperplane orthogonal to the 0th dimension, all nodes have a second connection to buses spanning the same dimension. The second bus direction differs from hyperplane to hyperplane but repeats if the width w of a dimension exceeds $D-2$. Figure 1 shows a three-dimensional dual-bus hypercube with width 3; the vertical direction is the distinguished dimension.

The torus is identical to the spanning bus hypercube except that it replaces the bus connecting each group of w nodes with a ring of point-to-point connections, improving the performance at an associated cost. As we shall see, the torus is a generalization of the binary hypercube used in the current generation of multicomputer networks. The performance motivations for this choice will become clear shortly.

Generalized hypercubes use a mixed radix numbering to obtain networks containing any number of nodes, in contrast to the limited expansion increments of traditional hypercubes.

The cube-connected cycles was also proposed to limit the number of connections to each node. This network replaces each node of a D-dimensional cube (i.e., of a binary hypercube) with a ring of D nodes. Each ring node connects to one of the D links incident on the vertex, fixing the node connectivity at three.

The bus-connected lens network proposed by Finkel and Solomon is created using a recursive construction algorithm.[6] In this network, an n-level lens contains $n(n-1)^n$ nodes, each connected to b buses.

The R-ary N-cube is a generalization of the indirect binary n-cube originally proposed by Pease.[13] In this network, N is the number of levels and R^N is the number of nodes on each level. The network contains NR^N nodes, each connected to $2R$ other nodes. Unlike Pease's original proposal, the cube is closed (i.e., the top and bottom rows shown in Figure I are the same).

An X-tree is a simple binary tree with all nodes at each level connected in a ring, reducing the communication bottleneck near the tree root.

Each of these networks can be analyzed to determine its network diameter and mean internode distance. These analyses are detailed, lengthy, and somewhat tedious. (The interested reader can find details in Reed and Fujimoto.[11]) In spite of this, these analyses have intrinsic value for two reasons. First, they represent the application of a uniform methodology to several networks. Second, many of the techniques embodied in the analyses are generally applicable to other networks and will likely be helpful when analyzing networks emerging in the future. To illustrate their application, we consider the torus, a generalization of the binary hypercube. Following this example, we show how the results of the analysis can be used to predict performance.

Analyzing the torus: An example

A D-dimensional torus—a D-dimensional lattice of width w—connects each of its w^D nodes to a ring of size w in each of the D orthogonal dimensions (see Figure 1). As its name implies, this interconnection network is topologically equivalent to a torus constructed in D-space. Because each of the w^D nodes is connected to D rings, there are $2Dw^D$ total link connections to the Dw^D communication links.

Message routing is easy if source and destination addresses are viewed as D-digit, base w numbers. Each digit represents a ring of w nodes, and a message is routed to its destination by successively sending the message to the correct location on each of the D rings. Consequently, the network diameter is just $D\lfloor w/2 \rfloor$, D times the maximum internode distance in any dimension using that dimension's ring. ($\lfloor x \rfloor$ denotes the integer portion of x—i.e., the largest integer less than x.)

When $w = 2$, this reduces to D, the binary hypercube case discussed earlier. If w is fixed and D increases, the network diameter increases only logarithmically with the number of nodes. This is the primary advantage of binary hypercubes. The offsetting disadvantage is that the number of connections per node also must increase with the network size. This means that the maximum network size must be predetermined, when the nodes are designed.

It initially appears that for a torus the mean internode distance under the uniform message routing assumption is also simply D times that for a simple ring. However, this is incorrect. In the torus, some dimensions need no address resolution; that is, the source and destination nodes share the same address on that ring. For these cases, the approach overestimates the mean path length. Instead, we must include the case of zero moves in each dimension and scale the sum of the moves in each dimension to obtain the true mean path length. Thus, the average distance moved in each dimension is

$$LV^{uniform}_{one\ dimension} = \frac{\sum_{k=0}^{w-1} \min\left\{ k, w-k \right\}}{w}$$

$$= \frac{w^2 - 1}{4w} \quad w\ odd$$

The minimum function in the sum reflects the routing of messages along the shorter of the two potential paths in the ring. Because the dimensions are independent, the true mean path length is D times this distance scaled to exclude nodes routing messages to themselves:

$$LV^{uniform}_{torus} = D \left(\frac{w^D}{w^D - 1} \right) LV^{uniform}_{one\ dimension}$$

Network symmetry and the existence of only one type of communication link allow us to immediately obtain the link visit ratios:

$$V^{uniform}_{CL} = \frac{LV^{uniform}_{torus}}{Dw^D}$$

$$= \frac{w^2 - 1}{4w(w^D - 1)} \quad w\ odd \qquad (4)$$

The derivation of mean path lengths and visit ratios for nonuniform message routing distributions, though conceptually straightforward, is computationally difficult. To simplify exposition, we consider only the case in which w is odd. The motivation for this is simple: for w odd, we can, without loss of generality, assume the source node is at the center of a D-dimensional hyperspace. Specifically, we can assume it is at the center of a $(D-1)$-dimensional hyperplane and has $\lfloor w/2 \rfloor$ $(D-1)$-dimensional hyperplanes "above" and "below" it. Suppose a message is sent to the l links above or below the hyperplane containing the source node, and the message is permitted to cross a maximum of L links. Then only those nodes within $L-l$ links of the point of origin in the destination hyperplane can be reached. This view of message routing leads to the recurrence

$$Reach(L,D,w) = \begin{cases} 1 & L = 0 \\ 2 & D = 1\ and\ 0 < L \leq \left\lfloor \frac{w}{2} \right\rfloor \\ 0 & D = 1\ and\ L > \left\lfloor \frac{w}{2} \right\rfloor \\ 2 \sum_{l=1}^{\min\left\{L, \left\lfloor \frac{w}{2} \right\rfloor\right\}} Reach(L-l, D-1, w) & \\ \text{otherwise} & \\ \quad + Reach(L, D-1, w) & \end{cases}$$

where $Reach\ (L,\ D,\ w)$ is the number of nodes exactly L links from a source node in a D-dimensional torus of width w. The first term in the last equation above contains a factor of two because a message can, by symmetry, go "up" or "down" from the source node. The second term in the last equation is the number of nodes reachable in the hyperplane in which the source node lies. This recurrence can be easily solved and a closed form for $Reach\ (L,\ D,\ w)$ can be obtained (though the symbol manipulation is arduous). Given the closed form, one can then easily obtain the number of nodes in the sphere of locality, the mean internode distance, and the link visit ratios.

Similarly, given the network diameter $D\ \lfloor w/2 \rfloor$, one can easily obtain the mean message-path-length visit ratios for the decaying probability message routing distribution.

Finally, the torus, with its lattice structure, has expansion increments of $w^D\ (w-1)$ when D, the dimension, is increased and $(w+1)^D - w^D$ when w, the width, is increased. Because expanding the number of dimensions requires two additional link connections per node, it will usually be necessary to rewire the entire interconnection network if this method of expansion is chosen.

Network performance bounds

In the formulation of message routing we have described, a message leaves a

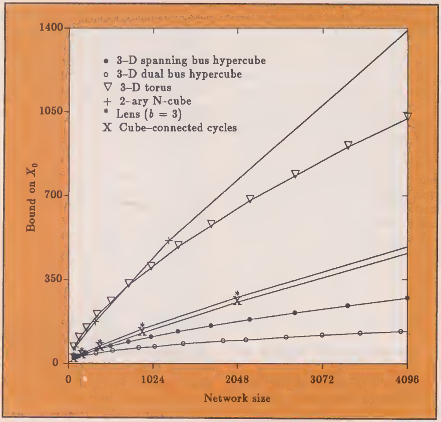


Figure 2. Upper bound on message completion rate with uniform routing.

source node, crosses some communication links to reach its destination, and causes some computation to take place there. After normalization, the visit ratio V_i can be interpreted as the average number of visits made to device i, a node or link, by a message. Suppose we define S_i as the average amount of service required by a message during each visit to device i. The product V_iS_i is the total amount of service required by an average message at device i, and the sum

$$\sum_i V_i S_i$$

is the total amount of service required by an average message at all nodes and links.

If the average number of messages circulating in the network is steadily increased, the utilization of at least one device must approach unity. Which device will saturate first? Because V_iS_i represents the average amount of service required at device i, the device with the maximum value of V_iS_i will first limit the message circulation rate. If X_0 denotes the rate at which a network can route messages from source to destination, an absolute upper bound on X_0 is given by

$$X_0 \leq \frac{1}{V_b S_b}$$

where

$$V_b S_b = \max_i V_i S_i$$

This technique, called asymptotic or bottleneck analysis in its most general form,[10] applies to any closed network in a steady state (that is, to any closed network in which the message arrival rate at each device equals the message departure rate). No assumptions about service time or queueing distributions are necessary. This simplicity allows one to make minimal assumptions about network behavior and, consequently, leads to conclusions applicable to a wide range of intended network environments.

To simplify comparison of networks, we assume that the mean time between message transmissions, S_{PE}, is the same at all nodes, and that all links require time S_{CL} to transmit an average message. We emphasize that this assumption is not required. The succeeding discussion can be applied in its entirety, albeit involving somewhat more arduous symbol manipulation, if each device has a distinct service time.

If there are T distinct link visit ratios, the bound on X_0, the network message completion rate, is

$$X_0 < \frac{1}{\max\{V_{PE}S_{PE}, V_{CL}^1 S_{CL}, \ldots, V_{CL}^T S_{CL}\}} \qquad (5)$$

Through analysis similar to that for the torus, we can derive values for V_{PE} and V_{CL} for many networks. Using these analyses, Figure 2 illustrates the bounds on the message completion rate for the uniform message routing distribution. Figure 3 shows the effect of locality on the message completion rate. In both cases, unit node and link service times are assumed. However, other parameters can be used.

In the uniform routing case, the torus and R-ary N-cube have the highest message completion rate, followed by the lens, spanning bus hypercube, and cube-connected cycles. A comparison of the mean internode distances for the same networks is decidedly different (Figure 4). Although there is a correlation between the mean internode distance and the bound on the message passing rate, it is not direct. The mean internode distance does not reflect the intensity of traffic on the links; the performance bounds do.

Although increased message routing locality obviously increases the message completion rate, the many different parameterizations of the locality distributions make it exceedingly difficult to draw conclusions without knowledge of some specific application area. However, constraints on the regions of feasible computation can be obtained.

Feasible computation quanta. In addition to providing bounds on the maximum rate of message transfer, the visit ratios can also be used to determine appropriate granules of computation, given the relative speeds of processors and communication links.

If K is the number of network nodes, the amount of service required at an average node is $V_{PE}S_{PE} = S_{PE}/K$. A linear increase in the message completion rate can be expected only if the communication links' VS products are no larger than this value. That is, increased parallelism is effective only if communication delays do not dominate. As prior figures have shown, the message completion rate is not, in general, such a linear function, implying that communication delays are limiting the message completion rate. Other than changing the message routing distribution, one can only adjust the ratio S_{PE}/S_{CL} to ensure that communication is not the performance-limiting factor. Inspecting Equation 5 shows that the minimum ratio of computation to communication, where communication delays are not dominant, is

$$\frac{S_{PE}}{S_{CL}} = \frac{\max\{V^1_{CL},\ldots,V^T_{CL}\}}{V_{PE}}$$

$$= K \cdot \max\{V^1_{CL},\ldots,V^T_{CL}\} \qquad (6)$$

In essence, the ratio of computation time to communication time for a message must be at least K times the maximum link visit ratio if the maximum computation rate is not to be limited by communication delays.

As an example, consider the torus again. The communication link visit ratios for uniform message routing are given by Equation 4, and the number of nodes is w^D. Applying Equation 6 yields

$$\frac{S_{PE}}{S_{CL}} = \frac{w^{D-1}(w^2-1)}{4(w^D-1)} \approx \frac{w}{4}$$

If communication delays are not to dominate performance, the ratio of computation time to communication time must increase linearly with w if the dimension of the torus is fixed. Simply put, larger granules of computation are needed for a wider torus. Conversely, increasing the dimension of the torus D does not require any increase in the ratio. This is the performance motivation for binary hypercubes: w is fixed at 2, only the dimension D grows, and the granule of computation is constant.

The lesson for designers of parallel algorithms is immediate and striking: the smallest feasible quantum of parallelism is dictated by the communication patterns of the algorithm *and* the network topology. At best, excessive parallelism leads to negligible performance gains. At worst, performance can decrease due to increased overhead.

Network selection. Situations do arise in which factors intrinsic to an intended application dictate the use of a specific ratio of computation to communication time for messages. When this occurs, the

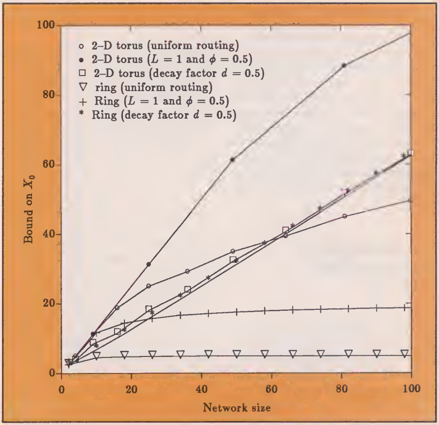


Figure 3. Upper bound on message completion rate with locality.

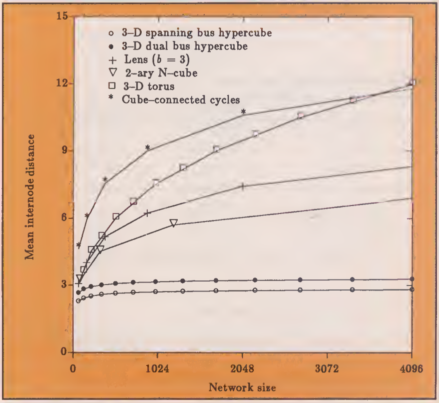


Figure 4. Mean internode distance with uniform routing.

designer must realize that the range of optimality for a specific network does *not* span the entire spectrum of network sizes. This is clearly illustrated by the crossing of the bounds on message completion rate in Figure 3. Thus, one may be justified in using one interconnection for ten nodes and a different one for one hundred nodes. One can analytically or numerically determine where the bounds for two networks cross by equating the *VS* products for the limiting devices, either nodes or links, and solving for the network size where they are equal.

Benchmarking multicomputer networks

Analytic models can capture the average, steady-state behavior of multicomputer networks, but they cannot easily reflect the time-varying behavior of real systems. Moreover, the idiosyncrasies of system software can be captured only through observation and measurement. For example, some commercial hypercubes such as the Ametek System/14 support only synchronous communication between directly connected nodes; others such as the Intel iPSC provide asynchronous transmission with routing. The effects of these differences are very difficult to model, but they can be measured easily.

Although analytic models cannot reflect all details of real systems, they do provide a framework for testing. For systems that support routing and asynchronous transmission, one can develop synthetic communication benchmarks that reflect communication patterns in both time *and* space. These benchmarks are based on the message routing distributions described earlier.

As analysis has just shown, binary hypercubes have attractive performance characteristics. To confirm this analysis and to highlight the interaction between analytic models and benchmarking, we examined the communication performance of two commercial hypercubes, the Intel iPSC and Ametek System/14. As shown in Table 1, both of these machines are based on the 16-bit Intel 80286 microprocessor. As noted earlier, the Intel iPSC uses Ethernet transceiver chips to manage internode communication. In contrast, the Ametek System/14 uses an Intel 80186 microprocessor to control internode communication via direct memory access.

Simple communication benchmarks. Data transfer time between two directly connected nodes is perhaps the simplest measure of communication network performance. Although independent of the type of interconnection network, this measure provides the value S_{CL} that is the basis for all network analysis.

If the messages transmitted between nodes are not broken into packets, message transmission time S_{CL} can be modeled as

$$S_{CL} = t_l + Nt_c \tag{7}$$

where t_l is the communication latency (i.e., the startup time for a transmission), t_c is the transmission time per byte, and N is the number of bytes in the message. Figure 5 shows the measured values of S_{CL} for both the Intel iPSC and Ametek System/14. Statistically, the linear model in Equation 7 is a good fit to experimental data on both hypercubes. Table 2 shows the result of a least-squares fit of the data to this linear model. The Ametek System/14 has a lower latency than the Intel iPSC, but the higher bandwidth of the Intel iPSC yields smaller total communication time for messages of more than approximately 150 bytes.

We emphasize that extrapolating general communication performance from single link tests is fallacious. Such simple tests highlight only link-level characteristics. More general benchmarks are needed to investigate routing and buffering behavior; this is the motivation for the synthetic communication benchmarks presented below.

Table 1. Hypercube hardware characteristics.

	Intel iPSC	Ametek S/14
Processor	8-MHz 80286	8-MHz 80286
Floating point	6-MHz 80287	8-MHz 80287
I/O procesor	none	10-MHz 80186
Minimum memory	0.5M bytes	1M byte
Maximum memory	4.5M bytes	1M byte
Channels per node	7	8
Peak bandwidth	10M bits/s	3M bits/s

Table 2. Hypercube communication comparison.

	Intel iPSC	Ametek System 14
Latency (seconds)	1.7×10^{-3}	5.5×10^{-4}
Transmission time (sec/byte)	2.83×10^{-6}	9.53×10^{-6}
Bandwidth (M bits/s)	2.8	0.84

Synthetic communication benchmarks. Application programs can be used to study hypercube communication performance, but because each program occupies only a small portion of the space of potential communication behaviors, a large number of application programs are needed to achieve adequate coverage. To study hypercube performance under a variety of communication traffic patterns, we have developed a model of communication behavior. Each hypercube node executes a copy of the model, generating network traffic that reflects some pattern of both *temporal* and *spatial* locality. Temporal locality defines the pattern of internode communication in time. An application with high temporal locality exhibits communication affinity among a subset of network nodes. However, these nodes need not be near one another in the network. Physical proximity is determined by spatial locality.

Temporal locality. Our implementation of temporal locality is based on a *least recently used stack model*, or LRUSM, originally used to study management schemes for paged virtual memory[14]; other models are possible. When used in the memory management context, the *stack* after memory reference $r(t)$ (i.e., the reference at time t) contains the n most recently accessed pages, ordered by decreasing recency of reference. The stack distance $d(t)$ associated with reference $r(t)$ is the position of $r(t)$ in the stack defined just after memory reference $r(t-1)$ occurs. These distances are assumed to be independent random variables such that *Probability* $[d(t) = i] = b_i$ for all t. For example, if $b_1 = 0.5$, there is a 50-percent chance that the next page referenced will be the same as the one just referenced.

In our adaptation of LRUSM to network traffic, each node has its own stack containing the n nodes that were most recently sent messages. In other words, destination nodes in the network are analogous to pages in the address space of a process for the LRUSM model of memory reference patterns. Parameters given to the model include the stack reference probabilities b_i and the stack size n.

We emphasize that the sum of the probabilities b_i is normally less than 1. Consequently, it is possible for the condition $d(t) > n$ to occur. In this case, a stack *miss* occurs. This corresponds to a page fault in the virtual memory context. In our model, a stack miss means that a new node not currently in the stack is chosen as the destination of the next transmission. This selection is based on a *spatial* locality model. This, in turn, is simply one of the message routing distributions discussed earlier: uniform, sphere of locality, or decreasing probability.

Experimental results. Figure 6 shows the mean time for a 16-node Intel iPSC to transmit 3000 messages whose length was drawn from a negative exponential distribution with a mean of 512 bytes. The horizontal line denotes the uniform message routing distribution. This provides a point of reference for the decreasing probability and sphere of locality routing distributions.

In Figure 6, $\phi = 1$ for the sphere of locality distribution. This means that all messages generated will be sent to destinations within the radius. Thus, the expected execution time approaches that of the uniform routing distribution as the radius

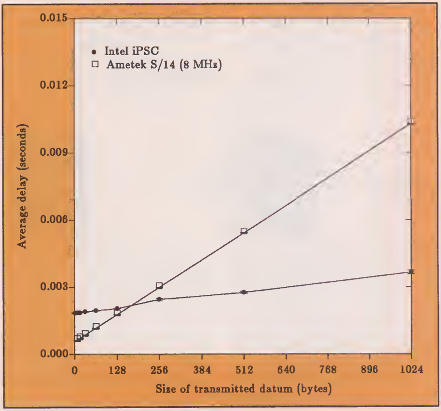


Figure 5. Hypercube data transfer benchmark.

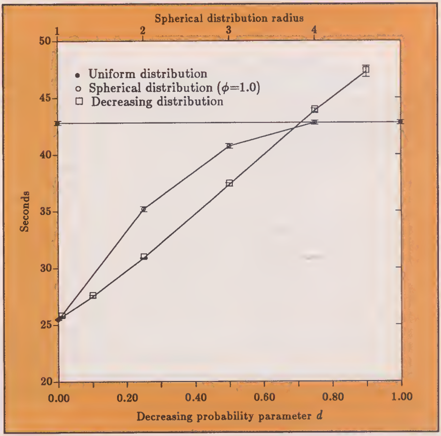


Figure 6. Spatial communication locality.

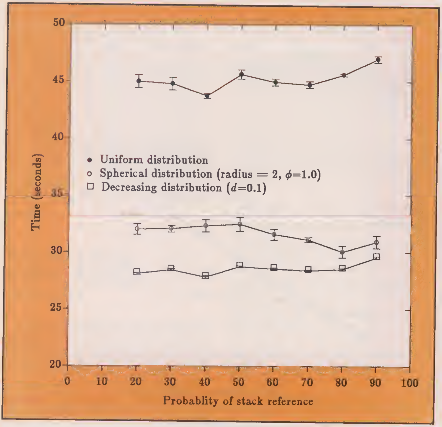


Figure 7. Temporal communication locality.

approaches the network diameter. With the decreasing probability distribution, increasing d means that a larger fraction of all messages are sent to distant nodes. For large enough values of d, the decreasing probability distribution is *anti-local.* Specifically, the mean internode distance is larger than that of the uniform routing distribution.

Figure 7 shows the effect of varying the temporal locality of the three spatial locality distributions, using the same number and type of messages as in Figure 6. In the figure, the temporal stack has a depth of one. This means that the number of messages sent to the node in the stack is the mean of a binomial distribution with parameter p, the probability of referencing the stack. Thus, there are "runs" of consecutive messages sent to a single destination node.

The most striking feature of Figure 7 is the small variation in the time needed to complete the suite of message transmissions. One would expect the temporal locality and its associated message runs to induce transient queues of outstanding messages on one link of each hypercube node. This phenomenon should manifest itself as increased delays as the probability of a stack reference increases. However, the variations shown in Figure 7 are not statistically significant. Why? Suppose the rate at which each node generated messages were perfectly balanced with the transmission capacity of each of the node's communications links. In this case, no message queues would form. Now suppose temporal locality were introduced. The presence of stack runs would induce an imbalance, a queue would form for at least one link, and messages would be delayed. This phenomenon does *not* occur in Figure 7, due to an imbalance of computation and communication speeds on the Intel iPSC. Nodes can generate messages faster than they can be transmitted by the communication links. Thus, queues develop for *all* links, independent of the presence of temporal locality, and these queues mask its effects.

Observations. Computation speeds, communication capacity, and communication patterns, both in time and space, interact in subtle ways. By using a synthetic benchmark that provides a broad spectrum of communication patterns, one can systematically and formally explore these interactions. This knowledge can be used to guide system design.

Could one analytically obtain the results shown in Figure 6? Yes, the analytic models do predict the observed behavior. However, predicting the specific crossing points and delay values requires knowledge of the node processing overhead and link transmission costs. Benchmarking is the only way to obtain these values. Thus, analysis and benchmarking are both important. One provides insight; the other provides concrete data.

Selecting an interconnection network suitable for all applications is difficult, if not impossible. However, a careful comparison of networks does show that there are classes of networks with ranges of performance. Among these, networks such as the torus and the R-ary N-cube seem most promising. As noted earlier, the binary hypercube is a special case of the torus.

When a network suitable for a specific class of applications is needed, a systematic approach based on a study of the most important applications in that class can expose the expected traffic patterns and computation requirements. Using these and the derived visit ratios, one can easily compare networks using asymptotic bounds.

A systematic approach is imperative. With such an approach, one can effectively compare new network topologies to existing ones and select networks suitable for specific application classes. □

> We think there remains much detail to be worked out, and possibly some further invention needed, before the design can be brought into a state in which it would be possible to judge whether it would really so work.
>
> —Committee examining Babbage's Analytical Engine

Acknowledgments

This work was supported in part under NSF Grant Number DCR 84-17948 and NASA Contract Number NAG-1-613 and by the Jet Propulsion Laboratory.

References

1. K.E. Batcher, "MPP—A Massively Parallel Processor," *Proc. Int'l Conf. Parallel Processing*, Computer Society Press, Washington, D.C., Aug. 1979, p. 249.
2. D. Hillis, *The Connection Machine*, MIT Press, Cambridge, Mass., 1985.
3. B.W. Arden and H.Lee, "Analysis of Chordal Ring Network," *IEEE Trans. Computers*, Apr. 1981, pp. 291-295.
4. L.N. Bhuyan and D.P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network," *IEEE Trans. Computers*, Apr. 1984, pp. 323-333.
5. A.M. Despain and D.A. Patterson, "X-TREE: A Tree Structured Multiprocessor Computer Architecture," *Proc. Fifth Ann. Symp. Computer Architecture*, in *ACM Sigarch Newsletter*, Apr. 1978, pp. 144-151.
6. R.A. Finkel and M.H. Solomon, "The Lens Interconnection Strategy," *IEEE Trans. Computers*, Dec. 1981, pp. 960-965.
7. L.D. Wittie, "Communication Structures for Large Multimicrocomputer Systems," *IEEE Trans. Computers*, Apr. 1981, pp. 264-273.
8. J. Rattner, "Concurrent Processing: A New Direction in Scientific Computing," *AFIPS Conf. Proc.*, Vol. 54, 1985 NCC, pp. 157-166.
9. M.T. Heath, *Hypercube Multiprocessors 1986*, Society for Industrial and Applied Mathematics, Philidelphia, 1986.
10. P.J. Denning and J.P. Buzen, "The Operational Analysis of Queueing Network Models," *Computing Surveys*, Sept. 1978.
11. D.A. Reed and R.M. Fujimoto, *Multicomputer Networks: Message-Based Parallel Processing*, MIT Press, Cambridge, Mass., to appear in Sept. 1987.
12. F.P. Preparata and J. Vuillemin, "The Cube-Connected Cycles: A Versatile Network for Parallel Computation," *Comm. ACM*, May 1981, pp. 300-309.
13. M.C. Pease, "The Indirect Binary *n*-cube Microprocessor Array," *IEEE Trans. Computers*, May 1977, pp. 458-473.
14. P.J. Denning, "Working Sets Past and Present," *IEEE Trans. Software Engineering*, Jan. 1980, pp. 64-84.

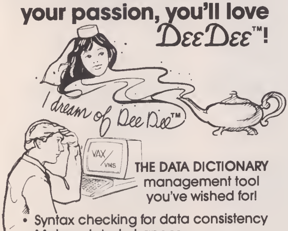
Daniel A. Reed is an assistant professor in the Department of Computer Science at the University of Illinois at Urbana-Champaign, where he was the recipient of an NSF Presidential Young Investigator Award. He also holds an appointment in the Center for Supercomputing Research and Development, where he is participating in the design of Faust, a programming environment for the Cedar shared memory multiprocessor. His interests include the design of parallel algorithms and software for message-based parallel processors. He is currently directing the design of Picasso, a hypercube operating system.

Reed received his BS, summa cum laude, in computer science from the University of Missouri—Rolla in 1978 and his MS and PhD, also in computer science, from Purdue University in 1983.

Dirk C. Grunwald is currently completing the requirements for the PhD degree at the University of Illinois at Urbana-Champaign. His interests are in multicomputer systems, operating systems and concurrent languages for ensemble machines, compiler design for vector and multicomputer systems, and concurrent algorithms. He received his BS and MS in 1983 and 1985 from the University of Illinois. He has been a summer intern with IBM-Boca Raton, DEC's Eastern Research Lab, and the hypercube concurrent computation project at the Jet Propulsion Laboratory. He is a member of IEEE and ACM.

Readers may write Reed at the Department of Computer Science, University of Illinois, 1304 West Springfield Ave., Urbana, IL 61801.

If STRUCTURED ANALYSIS is your passion, you'll love DeeDee™!



I dream of Dee Dee™
VAX/VMS

THE DATA DICTIONARY management tool you've wished for!

- Syntax checking for data consistency
- Makes global changes
- Alphabetizes automatically
- Menu driven and configurable
- Tailored automated documentation
- Automated creation of sub-definitions
- VAX EDT-like editing of data structures

Keeps **LARGE PROJECTS** under budget!
Powerful, Productive, Fast!
DeeDee ™ . . . $2995
Ada and Pascal type declarations
created with MkT™ **option . . . $995**

SCS inc. P.O. Box 15367 FORT WAYNE, IN 46885 219/432-3975

Reader Service Number 4

DMS SUPPORTS SOFTWARE ENGINEERS

Ultraware has announced the development management system DMS on DEC VAX/VMS and IBM PC/PS DOS, UNIX version to be announced. DMS is a software environment driver that integrates Meta-Data and execution system. DMS supports software development, configuration management and project management for the complete life-cycle. DMS features AI Analysis of the software process: solves the open shop SCM problem; traces development to project planning; maximizes management visibility; automates version control and build: supports SQA and IVV and T standards: traces test cases to requirements: computerizes the TRW unit development folder concept: helps human communications, and data sharing: controls access for work safety: provides callable interface library for customized environments. DMS goal is to support software development In a distributed heterogeneous operating system environment. DMS IBM PC/PS site license costs $10,000. Request IBM PC demonstration diskette without cost or obligation. **Contact Ultraware Ltd., P.O. Box 367, Kiryat Ono 55102, Israel, Phone: 972-3-341366, Telex: 4900004722 USA.**

Reader Service Number 5

CSM
'87

Advance Program

CONFERENCE ON SOFTWARE MAINTENANCE-1987

AUSTIN, TEXAS • SEPTEMBER 21-24, 1987

The Conference on Software Maintenance-1987 (CSM-87) will gather software managers, developers, maintainers, and researchers to discuss new solutions to the continuing challenge of software maintenance and software maintainability. CSM-87 will acquaint managers and practitioners with current advances and researchers with current needs.

SPONSORS:

Computer Society of the IEEE–
Technical Committee on Software Engineering

The Institute of Electrical and Electronics Engineers, Inc.

Data Processing Management
Association (DPMA)

National Bureau of Standards (NBS)

IN COOPERATION WITH:

acm ACM/SIGSOFT–
Special Interest Group on Software Engineering

Association for Women in Computing (AWC)

We Support the Future! Software Maintenance Association (SMA)

CONFERENCE COMMITTEE

GENERAL CHAIR
Roger J. Martin
National Bureau
of Standards
Gaithersburg, MD 20899
301-975-3295

PROGRAM CHAIRS
Dr. Robert S. Arnold
Software Productivity Consortium
1880 Campus Commons
Drive, North
Reston, VA 22091
703-391-1898

Wilma M. Osborne
National Bureau
of Standards
Gaithersburg, MD 20899
301-975-3339

TUTORIAL CHAIR
Dr. Mel Colter
Colter Enterprises
Colorado Springs, CO

TOOL DEMONSTRATIONS CHAIR
Gary Richardson
Texaco, Inc.
Houston, TX

PUBLICITY CHAIR
Mary Anne Overman
DOD
Ft. Meade, MD

LOCAL ARRANGEMENTS CHAIR
Dr. Brian Fugate
MCC
Austin, TX

IMMEDIATE PAST GENERAL CHAIR
Nicholas Zvegintzov
Software Maintenance
News
Staten Island, NY

PROGRAM COMMITTEE

Prof. David Bellin
Pratt Institute

Mr. Bruce Blum
Johns Hopkins University

Dr. Mel Colter
Colter Enterprises

Mr. Taz Daughtrey
Babcock & Wilcox

Prof. David A. Gustafson
Kansas State University

Mr. Tobey B. Marzouk
Marzouk & Parry

Mr. Steve Oxman
OXKO Corporation

Mr. Donald Parker
NASA-GSFC

Dr. David E. Peercy
BDM

Dr. Dieter Rombach
University of Maryland

Mr. Gary C. Sackett
Hughes Aircraft Co.

Prof. Norman Schneidewind
Naval Postgraduate School

Ms. Dolores R. Wallace
National Bureau of Standards

THE COMPUTER SOCIETY
OF THE IEEE

THE INSTITUTE OF ELECTRICAL
AND ELECTRONICS ENGINEERS, INC
IEEE

MONDAY, SEPTEMBER 21 • ALL DAY TUTORIALS

7:30	Registration
9:00-5:00	**Tutorials** I. Tutorial on Software Restructuring: Robert Arnold, Software Productivity Consortium (SPC) II. Advanced Practical Management Methods and Technologies for Controlling Software Maintenance Cost, Productivity and Maintainability: Tom Gilb, Consultant III. C, UNIX and Software Portablity: Alan Filipski, GTE and Arizona State

TUESDAY, SEPTEMBER 22 • GENERAL SESSION

7:30	Registration	
8:30	Opening Remarks	
9:00-10:00	**Plenary Keynote Address:** Dr. Barry Boehm, TRW, Improving Software Maintenance Productivity	
10:00	BREAK	
	TRACK A	**TRACK B**
10:30	**Session 1A–Panel** Metrics for Software Maintenance Management: James McCall, SAIC–Chair Panelists: Gene Long–SM-ALC, Major T. Shumskas–AFSC, Joe Cavano–RADC, Norm Schneidewind–NPGS, Bill Curtis–MCC.	**Session 1B–** Management Techniques and Procedures The Job of Software Maintenance: Ned Chapin, Infosci Building and Sustaining Software Maintainability: Wilma Osborne, NBS Improving Software Acquisition to Facilitate Software Maintenance: Robert Arnold, SPC
12:00	LUNCH	
1:30	**Session 2A** Empirical Studies Applying Control-Flow Metrics to COBOL: Linda Spratt, Barry McQuilken, Language Technology Evolution with System Sculpture: Some Empirical Results: Bruce Blum, John Hopkins University A Model for Software Maintenance: Dave Gustafson, Austin Melton, Kansas State University	**Session 2B–Panel** Legal Implications of Software Maintenance: Tobey Marzouk, Esq.–Chair Panelists: Pam Samuelson, University of Pittsburgh Ann Martin, SEI
3:00	BREAK	
3:30	**Session 3A-3B–Plenary Panel Session** Employing Standards to Improve Software Maintenance: Chair: Jerry Raveling–DOD, Wilma Osborne–NBS, Leonard Tripp–IEEE/Boeing, Bill Wilson–NASA	

WEDNESDAY, SEPTEMBER 23 • GENERAL SESSION

7:30	Registration	
8:30	**Plenary Panel Session:** James McCall, SAIC, Chair Embedded Computer Software Maintenance Environments Panelists: J. Patterson–SM-ALC, M. Parrish–OC-ALC, G. Myers–NOSC, D. Cerino–RADC, P. Pierce–SAIC	
10:00	BREAK	
10:30	**Session** 4A–Mini-Tutorial Reducing Risks in Software Maintenance David Peercy, BDM	**Session 4B**–Software Restructuring and Portability Software Recycling: Harry Sneed, Gabor Jandrasics, Software Engineering Service Maintenance and Reverse–Engineering–Low Design Documents Production and Improvement: P. Antonni, P. Benedusi, G. Cantone, A. Cimitile, University of Naples Portability Issues in Software Maintenance: Richard Head, AMOCO
12:00	LUNCH	
1:30	**Session 5A**–Advanced Techniques for Software Maintenance Workspaces and Experimental Databases–Automated Support for Software Maintenance: Dewayne Perry–AT&T Bell Labs Software Maintenance–Using Meta-Programming. Brian Terry, Robert D. Cameron, Simon Fraser University An Expert System Approach to a Program's Information/ Maintenance System: F Cross, Rockwell Collins Inc.	**Session 5B**–Mini-Tutorial: Control It or Lose It: Responsive Configuration Management (CM)–A Must for Software Maintenance: Jerry Raveling, UNISYS
3:00	BREAK	
3:30	**Session 6A–Panel** Status and Problems in Software Maintenance Education: Dave Gustafson, Kansas State University, Chair Panelists: Gary Sackett, Hughes Aircraft, Martha Walsh, AT&T Communication & Information Systems	**Session 6B**–Quantitative Evaluation of Maintenance Quantitative Assessment of Software Maintenance: H. Dieter Rombach, Victor Basili, University of Maryland Managing Software Cost and Quality: David Card, D.V. Cotnoir, C E. Goorevich, CSC Informations Systems Development Audits and Software Maintenance: Chung-Fern Wu, UCLA
7:00	CSM-87 Reception	

THURSDAY, SEPTEMBER 24 • GENERAL SESSION

8:30	**Session 7A** Formal Studies A Statistical Rationale for Evolution Dynamics Concepts: C.K.S. Chong Hok Yuen, C.M.R. de S-J Improved Domain Strategies for Detecting Path Selection Errors: Chen-Shing Chou, Min-Wen Du, National Chiao Tung University	**Session 7B**– Software Maintenance Tools A Source Code Analyzer for Maintenance: D. Richard Kuhn, NBS A Documentation Method Based on Cross Referencing: John Foster, British Telecom
9:30	BREAK	
10:00	**Plenary Panel Session:** Norman Schneidewind, Naval Post-Graduate School–Chair. Perspectives on Software Maintenance Panelists: Larry Druffel, SEI, Peter Freeman, UC/Irvine, Barry Boehm, TRW, Les Belady, MCC	
11:4S	Conclusions	

Tool Demonstrations • September 22-24
Continuous Demonstrations of tools by 2S Software Tool vendors.

Conference on Software Maintenance–1989 (CSM-89) Planning Meeting
S:30 p.m., Tuesday, September 22

Conference on Software Maintenance–1987
September 21-24, 1987
Austin, Texas

Complete and return this form along with your check to: CSM'87
c/o Computer Society
1730 Massachusetts Ave. NW
Washington, DC 20036-1903
Telephone: (202) 371-1013
Telex: 7108250437 IEEECOMPSO

Circle appropriate fee:

	Advance Registration Prior to 9/4/87	
	Member	Non-Member
Conference	$160	$200
Tutorial	$160	$200
	Late Registration After 9/4/87	
	Member	Non-Member
Conference	$190	$240
Tutorial	$190	$240

☐ Student Conference Fee: $50
☐ Check Enclosed
☐ VISA ☐ MasterCard
☐ American Express ☐ Choice
Total Enclosed: $____________________.

Card No __________ Exp Date __________
Signature __________
Name __________
Affiliation __________
Mailing Address __________
City/State/Zip __________
Work Phone __________ Home Phone __________

Registration Fee includes proceedings and reception Request for refunds must be received in writing no later than September 4, 1987

La Mansión
Hotel Reservation Form
Conference on Software Maintenance–1987
September 21-24, 1987

Please check accommodations and price desired:
☐ Single occupancy: $74.00
☐ Double occupancy: $84.00
Arrival Date __________ Departure Date __________
☐ am ☐ pm ☐ am ☐ pm

Reservations must be received by the La Mansión by September 1, 1987.

Reservations received after September 1, 1987 will be accepted on a space available basis.

Complete this form and return to:
La Mansión Hotel
Reservations
6505 IH-35 North
Austin, Texas 78752
(512) 454-3737

All major credit cards accepted:
☐ VISA ☐ MasterCard
☐ American Express

Card No __________
Exp. Date __________
Signature __________
Name __________
Address __________
City/State/Zip/Country __________

To confirm a reservation, a one room's night advance deposit is required.
Check in 3:00 p.m. Check out 12:00 p.m.

"Any clod can have the facts,
but having opinions is an art."
Charles McCabe
San Francisco Chronicle

THE OPEN CHANNEL

Jim Haynes, Computer Center, US Santa Cruz, CA 95065
Bitnet, haynes@ucscc; Arpa, ucscc!haynes@ucbvax.berkeley,edu

My personal profiles of programmer personnel

Programmers span a wide range of attitudes, abilities, experience, and education. Their backgrounds influence their perception and reception of new ideas and tools. This article proposes a scientific scheme for classifying programmer personnel. It is based on my personal, o*pun*-minded observations (and years of frustration) from dealing with software users and developers. The recognition of the programmer class or category to which these individuals belong can prove useful in developing tools and documentation, as well as in communicating with them in general.

Programmer classifications. Programmers generally fall (or are pushed or shoved) into one of four classes:

- Novices (New, Overzealous, Very Inquisitive ComputEr Students),
- Wimps* (Well-Intentioned, Mediocre ProgrammerS),
- Pros (Perceivably Reliable, Omnipotent Software Engineers), and
- Prima Donnas (PeRmanently IMmutAble Software Developers Of Notorious Narcissistic Attitude).

(Note: Some programmers have no class, in which case they don't fit in anywhere. Unfortunately, it is beyond the scope of this article to discuss the backgrounds of my relatives.)

Novices. Novices are usually fresh out of school, starry-eyed, and easily motivated. They are very receptive to "playing with" new tools or adopting new development methodologies, since they are still in learning mode. They have very little invested in previous techniques, and have yet to be burned by flakey software and by hideous compiler bugs. They have yet to be christened into the real world and put through the school of hard knocks.

Wimps. A majority, unfortunately, of the professional programmer community falls into this category. Battle scarred and war weary, they are leery of innovation. This group really needs to be sold on the technical merits of any new system or technique (that is, they need to be told what's in it for them). They are comfortable and reasonably productive with their old (possibly antiquated) tools and methodologies. Unless properly motivated or threatened, they will dismiss any efforts to extend their capabilities. The rhetorical question becomes, "Can you teach an old programmer new tricks?"

Pros. These are the technical gurus in any software organization—the people to whom the Novices and Wimps (but not the Prima Donnas) go for assistance. "Working smarter, not harder" is their motto, and they are always receptive to using or abusing any new tool or software system that they can buy, borrow, or steal. Pros do not need to be motivated, since they will rapidly latch onto any technology that they perceive as offering them leverage and enhancing their ability to perform their jobs.

Prima Donnas. Prima Donnas embody the antithesis of egoless programming. They refuse to accept any new tool or technology unless they can get credit for thinking of it themselves. (After all, they are legends in their own minds.) Also, Prima Donnas often lose touch with reality and develop systems that are either incomplete or totally useless to anyone but themselves. Motivating Prima Donnas to step down from their thrones is a management challenge that often boils down to a battle of wills (that is, "You will do it, or else!").

Postscript. Enabling people to make the transition from one software technology to another involves developing new tools, techniques, and associated training methods to facilitate the dissemination, assimilation, and eventual application of advances in the state of the art. The effectiveness of any approach is further enhanced when the backgrounds of the targeted individuals are taken into consideration, and the respective tools, techniques, and training methods are tailored accordingly. The classification scheme for programmer personnel proposed here addresses these issues in a *pun*gent but pragmatic perspective.

William J. Tracz
Computer Systems Laboratory
Stanford University

"The Open Channel" is exactly what the name implies: a forum for the free exchange of technical ideas. Try to hold your contributions to one page maximum in the final magazine format (about 1000 words).

We'll accept anything (short of libel or obscenity) so long as it's submitted by a member of the Computer Society. If it's really bizarre we may require you to get another member to cosponsor your item.

Send everything to Jim Haynes at the address given above.

---

*This classification was inspired by Bill Neugent's "Well Intentioned, but Mediocre People" category, which was discussed in "Preposterous Opinions About Computer Security," SOGSAC Review, Vol. 4, No. 3, Summer 1986.

Look before you leap to buy new technology

And let not technology blind thine eye!

If something like that verse isn't written somewhere in every software-engineering manager's "Operations Bible," it ought to be. The urge to use immediately the latest technological innovations lies deep within all of us. It's a carryover from the earlier days of data processing, when new technology was generally assessed in terms of quantity rather than type. Newer computers were faster than their predecesors, had larger memories, or were blessed with bigger on-line storage devices, but the basic arrangement was the same: centralized systems with communications tentacles to peripherals and customers.

That's not the case anymore. In the 1980's, new technology is assessed in terms of type as well as quantity. Faster processors, larger memories, and bigger storage devices for centralized systems continue to be announced, but whole new concepts of computer technology are also available—witness the advent of personal computers, computer workstations, and local area networks. The difficulty is that in the hurry to embrace these wonderful new tools, software-engineering management may forget that properly applying them is significantly different from just plugging in another larger CPU or a dumb terminal.

The dumb-terminal-replacement syndrome is particularly acute in cases of large, complex software-engineering applications. Such applications typically employ 20, 30, 40, or even more software engineers. The chains of management command and technical communication are long. Procedures and interactions for organizing and performing the work are complex. Even with the use of dumb terminals connected to a centralized system, integration of separately produced modules and components is difficult at best and can become totally impossible if careful design and control are not followed at every stage of the work. Adding personal computers or computer workstations connected by means of local area networks can easily exacerbate the difficulty of work on large, complex, software-engineering applications.

In a centralized system, all the modules and components needed for a large software-engineering application are located in a concentrated area comprising a few disk drives. Formal library structures and processes (for example, Librarian, Source Code Control System, Code Management System) can be readily established so as to control placement of, updates to, and relationships between cooperating entities. Tests of modules that are under development or from different sources are easy to coordinate. The software engineers merely adjust a series of pointers to each of the proper disk files and away the tests go.

With personal computers or computer workstations, work on large software-engineering applications becomes much more difficult. Copies of modules under development reside on disks belonging to physically separate computers. Coordination requires an explicit transfer of data from one machine to another, which is a slow-moving process when compared with changing pointers on a centralized disk file. Identifying and controlling the proper entity versions to ensure that the proper pieces are tested is a never-ending manual process. When the time comes to build the best-and-final, deliverable version of a system, significant manual effort may be required to locate and transmit all of the appropriate components into the designated configuration and assembly control point. Overall, instead of improving productivity in developing integrated software, the fragmenting of work and assigning of it to separate personal computers or computer workstations can end up degrading it.

The monetary costs of personal computers and computer workstations can also be less attractive than one initially perceives them to be. The typical large project is usually already equipped with at least a superminicomputer-class processor (for example, the DEC VAX, DG MVx000, IBM 43xx) that acts as a centralized configuration-control and general-automation machine at a cost of from $200,000 to $500,000. Most of today's computer workstations (for example, those produced by Apollo and Sun) cost in the neighborhood of $18,000 when they are equipped with their necessary complement of software. On a 40-person project, at least one workstation per two software engineers is desirable, which makes for a total cost of $360,000. If we assume that a dumb terminal costs $1000 per copy, one such terminal for every project participant would cost a total of only $40,000. The difference in the cost of software-engineering terminals is thus $320,000, which is enough to increment a project's initial superminicomputer (for example, its processor, memory, and/or on-line storage) significantly, or even to purchase another supermini. As a consequence, for a large software-engineering project, the computing-horsepower-versus-cost argument currently used to justify computer-workstation acquisition can become highly questionable.

Substituting a $3000-per-copy personal computer for a workstation decreases the difference in cost. Unfortunately, a typical personal computer has nowhere near the calculation or storage capacities available on the current crop of computer workstations, and as a result, it may not be suitable on large projects.

In those cases in which the component-fragmentation problem and the cost difference between computer workstations and personal computers represent nonsignificant issues, two other aspects should be considered before taking any plunge: machine operations and data integrity/backup.

Machine operations involve the tasks necessary to bring any computer to a usable state on a routine basis. For a computer workstation or personal computer, power-on and boot sequences are usually executed at least once each day. Disk formatting; changing printer ribbons and paper; cleaning screens, keyboards, and print heads are also necessary jobs that are required on a periodic but less frequent basis. In a centralized facility, one or two dedicated operations people generally perform the bulk of these tasks. In a dispersed-computing environment, such tasks tend to be left to the individual computer users, who are expensive software engineers. The intent of automation is to make valuable personnel, who are scarce, more productive. Performing routine computer operations is not usually considered one of the best uses of these folks' time.

The *data integrity/backup* aspect follows much the same procedure. Dedicated operations personnel perform controlled backup processing as a routine part of the operation of any centralized facility. In dispersed-computing environments, the user is left many times to do his or her own backups. Even if an aide is assigned to perform this function, the aide must do it once per operation for each workstation at each location, and he or she must do it on a noninterfering basis.

Neither a computer workstation nor a personal computer is the same as a dumb terminal. Before jumping to implement these latest qualitative improvements in computing technology, software-engineering managers must carefully weigh the associated benefits and costs. It's actually a relatively straightforward task. We've just got to keep our eyes open.

David A. Feinberg, CDP
Seattle, Wash.

STANDARDS

Editor: Helen M. Wood, National Bureau of Standards, 8154 Technology, Gaithersburg, MD 20899; (301) 975-3240.

Design Automation Standards: The Promise and the Dilemma

Ron Waxman, Chair,
Design Automation
Standards Subcommittee

Electronic system design and manufacturing require data covering all design and manufacturing process aspects, in addition to providing data for logistic support (maintenance and technology insertion) of the final product. Creating, modifying, supporting, and managing complex electronic systems is dependent upon the ability to communicate the design details from one design discipline to another and to integrate the various types of data representing the many facets of the product description.

The electronic design community utilizes the CAE/CAD tools and data representations oriented toward the description of system architecture and the description of logic. The technologists utilize data formats and tools to assist them in the design and description of detailed circuits and in the creation of metallization patterns that will implement the circuits. The mechanical designers create the package constraints to which the electronic designer must conform. A data communication problem exists within each "island" of design discipline and between those "islands." For example, all electronic parts end up in mechanical subassemblies and complete systems, resulting in some common data requirements. Presently, each activity uses different syntax and semantics. Common data syntax and semantics may potentially be achieved within each island of design discipline, but are more difficult to achieve between islands.

An attempt is being made to provide paths of communication among and within these various activities. Several independent standards-making activities are proceeding. To say that an attempt is being made to coordinate these activities is presumptuous. They are too large in and of their own right, and each has its own internal technical problems. Paths of communication are the most that can be hoped for (equivalent to communications between nations). However, the communications may result in a better understanding by each standards-making body of the objectives of others. This improved understanding may lead to less overlap and better means of integrating the use of each standard, as it unfolds.

The parallel standardization efforts for standards applicable to the area of design automation (DA) may give the impression of conflict in the DA standards arena. The contention of the Design Automation Standards Subcommittee (DASS) is that these emerging standards can work together harmoniously.

The Technical Committee on Design Automation, within the Computer Society of the IEEE, sponsors DASS, which has several working groups, each developing a standard, recommendations for standards, or recommended practices for using such standards, across the full range of the electronic design spectrum (i.e., product life cycle). VHSIC Hardware Descriptive Language (VHDL) standardization is one such activity. Balloting for a VHDL standard is now underway. The VHDL work is being done in cooperation with the Automatic Test Program Generation Subcommittee of the IEEE Standards Coordinating Committee #20, ATLAS. A new effort to develop Electronic Design Interchange Format (EDIF) as an IEEE standard is just getting started. This effort is being conducted by a DASS working group and is dependent on lines of communication to the ad hoc EDIF Committee, an independent organization. Another cooperative effort on conceptual data modeling for electrical products is being jointly sponsored by DASS and the Initial Graphics Exchange Specification (IGES) Electrical Applications Committee. DASS has established a working group to address this task.

The overall purpose of DASS is to provide standard description languages for the capture, utilization, and communication of electronic design data. The scope of the effort spans the creation of the ability to describe and model abstract notions of system design down to concrete representations of circuit and layout data for electronic systems. The descriptions are suitable for use as specifications and as data files to be used in combination with various design tools by designers in the process of achieving a given design. To ensure that the languages may be used effectively in the design process, some DASS activity is directed at the development of recommended practices for design management.

To accomplish this purpose, the DASS has established several working groups. In addition to those mentioned above, work is just starting or is ongoing in the following areas:

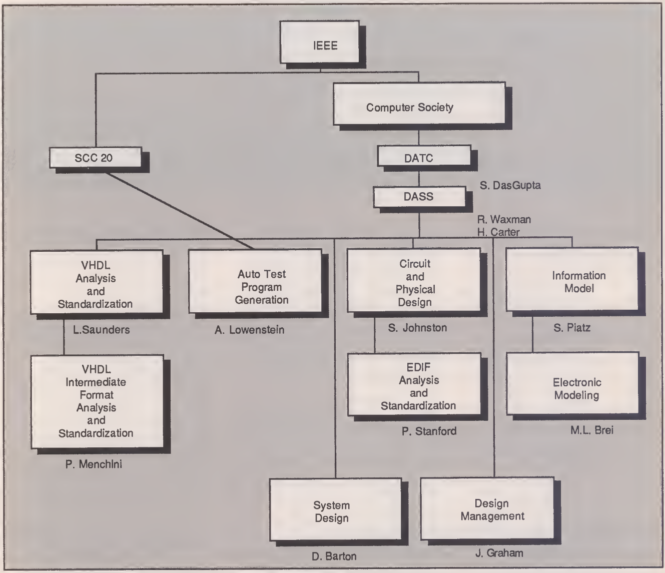
- system design language requirement generation,
- VHDL intermediate format standardization, and
- information modeling techniques and requirements for circuit and physical design descriptive data.

The scope and purpose of each of the working groups is described in the following paragraphs.

Working Group on Circuit and Physical Design (WGC&PD)
Chair: Susan Johnston; (206) 657-6658
Project title: A Standard Description Language for Computer Circuit and Physical Design
Scope of proposed standard: The standard is the description of the interconnection of logical gates and transistors

Why is there so much interest in standards? What are the issues—economic, technical, political—involved? Who are the players? These are among the topics that will be considered in the Standards Department.

Short articles on standards-related topics are sought for publication in this department. For details regarding scope and length, contact the department editor at the address given above.



DASS organization.

and the geometric representation of these interconnections and transistors.
Purpose of proposed standard: To provide a syntax, vocabulary, and semantics for describing electrical and physical data required to specify and manufacture semiconductor circuits and assemblies.

EDIF Analysis and Standardization Group (EASG)
Chair: Paul Stanford; (214) 997-2777

This is a newly formed group. Its mission is to standardize a version of EDIF that is acceptable to the IEEE participants. The description of the project under the WGC&PD applies to the EASG.

VHDL Analysis and Standardization Group (VASG)
Chair: Larry Saunders; (507) 253-2509
Project title: A Standard Description Language for Electronic Hardware
Scope of proposed standard: Abstract modeling levels to logical gate level of description.
Purpose of proposed standard: To provide a syntax, vocabulary, and semantics for describing system design data. The data is required to support the design, production, and maintenance of electronic hardware throughout its life cycle.

VHDL Intermediate Format Analysis and Standardization Group (VIFASG)
Chair: Paul Menchini; (919) 361-1913

This is a newly formed group. Its mission is to standardize an intermediate format compatible with VHDL that will provide a standard means for CAD tools to interface to VHDL.

Working Group on Information Model (WGIM)
Chair: Steve Piatz; (612) 456-4438
Project title: A Standard Description Language for a Common Computer Information Model

Scope of proposed standard: System- to circuit-level functional, electrical, and physical (geometric) design data in an internal machine-oriented form that is mappable from and to the external forms of the data provided by other standards developed by the DATC Design Automation Standards Subcommittee Working Groups, and other related standards.

Purpose of proposed standard: To provide a syntax, vocabulary and semantics for describing a standard internal representation of functional, electrical, and physical design data. The internal representation will map from the external forms of graphical or textual data to the internal form. The internal form will be used to drive design tools and manufacturing processes.

Electronic Modeling Group (EMG)
Chair: M. L. Brei; (703) 361-7050

This group is presently developing a model that will be applicable to the electronic design environment in general. They are investigating the application of the model, as it develops, to VHDL and EDIF.

Working Group on Design Management (WGDM)
Chair: John Graham; (301) 688-6877
Project title: Recommended Practices for Management of Computer System and Hardware Design
Scope of proposed standard: The recommended practices will address all elements required to manage digital computer system and hardware design, manufacture, and life-cycle support.
Purpose of proposed standard: To provide a document describing recommended practices for managing complex digital computer system and hardware design processes.

Working Group on System Design (WGSD)
Chair: Dave Barton; (301) 657-3775

This working group has just been reactivated. Its purpose is to develop requirements for a system design language. System design, in this context, relates to the area of abstract modeling of function and hardware/software trades.

A steering committee coordinates the activities of the working groups. Ron Waxman ([703] 367-2167) is chair. (After July 1, Ron Waxman can be reached at [804] 924-6097.) Hal Carter ([513] 475-3125) is cochair of DASS. Other members of the DASS Steering Committee, in addition to the working group chairs, are Gordon Adshead, Craig Winton, Steve Blumenau, Ron Short, Jwahar Bammi, Al Lowenstein, Vishwani Agrawal, John Hines, Victor Berman, Larry O'Connell, Moe Shahdad, and Manuel d'Abreu.

Many of those participating in the various DA standards activities are working hard to resolve issues and to provide industry with a set of Computer/DA Standards that make sense. The acknowledgement of the importance of complementary standards efforts by those who may be advocating a given standard as well as by others should help to confirm that the process is now starting to come together. Indeed, there will be conflict, but the formalization of paths of communication between standards-making groups will eventually resolve those conflicts. Each participant in the standards development process must come to recognize the total span of requirements for DA standards. Then we can reach agreement and productively address each evolving standard.

Success requires a flexible standards-making organization. The foundation has been laid for the ongoing work of DASS. DASS will build on this foundation during the coming year, produce one new standard, and produce some preliminary design automation standards material for perusal by the electronics design community. The standards support the industry activity, yet they also establish direction.

If the reader wishes to help formulate the future direction for design automation, he or she should become active in DASS. Contact the person(s) listed above appropriate to your interests.

The next DASS meeting and working group meetings will be held prior to the 24th Design Automation Conference in Miami Beach, on Saturday and Sunday, June 27 and 28, 1987.

New DIBOL Standard Expected to Stabilize Current Implementation

The DIBOL programming language is a general-purpose business programming language well-suited for writing financial applications for small business computers. It was originally introduced 14 years ago and underwent major feature enhancements in 1975 and 1983.

Because of the expanding use and the variety of implementations, the need has arisen for a clearly defined standard. The current implementations are similar, but incompatible. X3 has developed a draft proposed American National Standard to provide an unambiguous, machine- and operating system-independent definition of the DIBOL language. This draft standard (X3.165-198X) is available for public review and comment for a four-month period ending August 10, 1987. Copies may be obtained from Global Engineering Documents, Inc. by calling (800) 854-7179 for a single-copy price of $30.

International Standard for a Generalized Markup Language Published

The world now has an internationally accepted standard for a generalized markup language for programming computer-assisted text processors widely used by the publishing industry and companies that generate in-house publications. ISO International Standard 8879 is available in the United States from the American National Standards Institute.

The markup language specified in ISO 8879 makes processing easier and provides for flexibility in output by separating a document's text from the instructions that dictate style for such features as titles, subheadings, paragraphing, and lists. Separation is achieved by applying generic identifiers and generic coding to document elements requiring special treatment.

The new international standard includes an abstract syntax for markup of document elements. To accommodate the many environments in which the standard will be used, provision is made for customizing the language and taking into account different user preferences in text entry conventions and different requirements imposed by a variety of keyboards and displays.

Use of the standard is expected to reduce markup costs and production lead time and allow interchange among word processors with different functional capabilities. Another expected advantage is the ability to create auxiliary documents like mailing lists from original text.

The standard was developed by Technical Committee 97, Information Processing Systems, of the International Organization for Standardization (ISO). ANSI administers the TC 97 Secretariat.

Copies of ISO 8879-1986—Standard Generalized Markup Language (SGML)—are available from ANSI's International Sales Department ([212] 354-3300) at $50 each.

To speed up the development of SGML products and to validate their conformance to the standard, the National Bureau of Standards has developed an SGML reference implementation and a preliminary set of tests. For more information, contact: James Heath, Office Systems Engineering and Reliable Systems Group, Room B266, Technology Building, National Bureau of Standards, Gaithersburg, MD 20899; (301) 975-3350.

CALL FOR PAPERS, PANEL SESSIONS, & PARTICIPANTS

COMPSTAN '88

The Conference on Computer Standards

Sponsored by The Computer Society of the IEEE
In cooperation with the Technical Committees on
Microprocessors and Microcomputers
Computer Communications
Operating Systems
Software Engineering
Test Technology
Design Automation
and the Standards Co-ordinating Committee

DATE: March 21-23, 1988
WHERE: Sheraton National—Washington, D.C. Area
THEME: Computer Standards Evolution: Impact and Imperatives

Voluntary standards, government standards, and de facto industry standards affect the use and productivity of both hardware and software. COMPSTAN will focus on the evolution of computer standards, their impact on both users and producers, the effective utilization of computer standards, and some of the major emerging standards.

Scope and Purpose:

Computer standards are an important fact of life in the development and use of computers today. National and international computer standards can significantly impact both a producer's ability to deliver quality computer products, and the users' ability to obtain and use them.

COMPSTAN will provide a forum for the standards using community to discuss, 1) the *impact* of standards; and 2) the *imperatives* of what must be done to insure that standards are used to advance the computing state-of-the-art and state-of-the-practice, rather than as a stifling force against progress.

Important Dates:

Submission Deadline: September 14, 1987
Acceptance Notification: November 16, 1987
Conference: March 21-23, 1988

Suggested Topics:

Paper and panel discussion proposals related to, but not limited to, the following:

- Impact of Standards on Daily Practices
- Is Consistency between National & International Standards Important?
- Standards Development Process: Evolution or Revolution?
- Legal Issues: Anti-trust vs. Cooperative Efforts
- Legal Issues: Product Liability and Standards Development Liability
- Critiques of Emerging Standards
- Security & Standards
- Product Implications of Standards
- Implementing Standards: Successes & Failures
- Codifying the Past vs. Directing the Future
- Stabilizing Technology with Standards: Good or Bad?
- Testing Conformance to Standards
- Government Imposition vs. Voluntary Adoption
- Consortiums and Standards
- Who are the Benefactors of Standards?
- The Need for New Standards: Where? When?
- Do Standards Stifle or Promote Technology Innovations?
- Impact of Standards as an Economic Battleground
- Eliminating Standards Myths

Submission Information:

Papers: (5 copies double spaced) 1000-5000 words. Papers must not have been published or submitted elsewhere for publication. The cover letter must include: title and maximum 250 word abstract only. The first page must include title, all authors' names, complete mailing addresses, and telephone numbers. If the paper is accepted, one of the authors must agree to present the paper at COMPSTAN '88.

Panel Session Proposals: (5 copies) Panel session proposals must include: name of panel session organizer, mailing address, telephone number, panel topic's significance for COMPSTAN '88, and a list of 4-5 potential panelists.

Submit papers to program co-chair James A. Hall at address listed below.

Conference Committee

General Chair
Roger J. Martin
National Bureau of Standards
Gaithersburg, MD 20899

Program Co-chairs
Laurel V. Kaleda
IBM Corp.
L43/098
5600 Cottle Rd.
San Jose, CA 95193
(408) 284-3026

James A. Hall
National Bureau of Standards
Bldg. 225, Rm. A266
Gaithersburg, MD 20899
(301) 975-3273

Publicity Chair
Walter J. Ellis
IBM Federal Systems Division
6600 Rockledge Drive
Bethesda, MD 20817

Vice President for Standards
Helen M. Wood
National Bureau of Standards

Program Committee

Harold Carter, *University of Cincinnati*
Carl Friedlander, *BDM*
Tom Hannan, *CTA*
Charles Hollocker, *Northern Telecom*
John Horch, *Teledyne/Brown*
Jim Isaak, *DEC*
Tom Kurihara, *Dept. of Transportation*
Don Loughry, *Hewlett-Packard*
Perry Nuhn, *Software Productivity Consortium*
Robert Poston, *PEI*
Pete Prinzivalli, *PRD Systems*
Michael Smolin, *Strobe*

For further information contact: Computer Standards Conference, c/o Computer Society of the IEEE, 1730 Massachusetts Ave., N.W., Washington, D.C. 20036-1903, (202) 371-0101

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC
IEEE

THE COMPUTER SOCIETY OF THE IEEE

IEEE survey reports members' incomes

The average income of non-student US members of the IEEE was $53,889 as of January 1987, according to the *1987 IEEE U.S. Membership Salary and Fringe Benefit Survey*.

The survey, the eighth conducted by the IEEE's United States Activities Board, noted that IEEE members in Atlantic City, New Jersey, and West Palm Beach, Florida, received the highest average salaries, with means of $68,400 and $67,400, respectively. A special survey analysis concluded that members living in Santa Cruz, California, and Atlanta, Georgia, fared the best overall.

USAB's January survey was based on responses of around 13,000 randomly-selected non-student U.S. members engaged in electrical, electronics, and computer engineering, and computer sciences.

The survey measured salary by highest academic degree, experience, age, sex, primary area of technical competence, employer, job function, and company size. For the first time, salary was also presented according to metropolitan statistical area.

Generally, salaries rose with academic degrees obtained, increased levels of responsibility, and experience. However, on the average, women made less than men regardless of years of experience. For example, the mean income for men with 11 to 15 years of experience was $54,100, while that for women with comparable experience was $49,600.

The 107-page study was prepared by Number Crunchers of Takoma Park, Maryland, under contract to the USAB. The survey can be ordered for $55.95 (IEEE members) or $69.96 (nonmembers), with a handling charge of $5. Send prepaid orders for publication number UH0175-0 to the IEEE Service Center, 445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331. Credit card orders can be placed through Publications Sales, (201) 981-1393.

Algorithm replaces Fourier transform

Ronald Bracewell, a professor of electrical engineering at Stanford University, has patented an algorithm to replace the Fourier transform.

He named the algorithm after Ralph Hartley, because of Hartley's work at Bell Laboratories in 1942, but the Patent Office calls it the Bracewell Algorithm. According to Bracewell, the former name seems to have stuck among engineers.

Bracewell assigned the patent to the Trustees of Leland Stanford Junior University, which will license out the algorithm. According to the university, once the overhead is paid for administering the patent, Bracewell will earn one-third of the income from the licenses.

The inventor is reputedly trying to convince other engineers at Stanford to build a chip containing the algorithm, which would give the university a patent over both the algorithm and the chip.

Expert says women's gains slowing

Gains made by women in science and engineering during the past 15 years may be slowing and some advances of the past decade may not last, according to an expert on personnel in those fields.

Betty M. Vetter, executive director of the Washington-based Commission on Professionals in Science and Technology, tackled the subject in an article published in the latest edition of *Mosaic*, a magazine put out by the National Science Foundation.

According to Vetter, the educational climate for women has improved in the last 15 years and will probably continue to do so as universities try to attract good students from a shrinking college-age population. However, she expressed doubt that women will continue to expand their participation in science and engineering. One reason is that women continue to be more highly represented in the social and life sciences. Another is that opportunities in the marketplace continue to lag behind those for men. Moreover, several programs to recruit women into science and engineering have been dropped.

The commission Vetter directs is a private nonprofit organization formed in 1953 by major scientific societies to focus on problems in scientific manpower. It was formerly called the Scientific Manpower Commission.

Vetter's article in *Mosaic* is available free of charge while supplies last from the National Science Foundation, Public Affairs, Room 527, 1800 G St., Washington, DC 20550.

NSF establishes new computing research center

The National Science Foundation has chosen the University of Colorado at Boulder in cooperation with Colorado State University to receive $14.5 million over the next five years for the establishment and funding of an Engineering Research Center for Optoelectronic Computing Systems.

The goal for the new center is to create optoelectronic devices and systems for computing, signal processing, and artificial intelligence.

Thomas Cathey, professor of electrical engineering at the University of Colorado, will direct the center.

NSF's Division of Cross-Disciplinary Research, assisted by a liaison team, will oversee the center. DCR assesses the progress of each NSF-established center annually to determine funding levels for the following year. During the third year, a site visit panel will review the center. The evaluation will help determine whether NSF will continue funding the center for an additional five years, or phase it out over a two-year period.

NSF support terminates after a maximum of 11 years.

NSF expects the center to improve the United States' competitiveness in the communication, information, and manufacturing industries.

Substitute trade bill urged

The House Republican Policy Committee released an official statement April 29 that urged members to support a bipartisan substitute trade bill designed to toughen relations with US trading partners but without resorting to automatic retaliation.

According to the committee chair, Dick Cheney (R-Wyo.), the Democratic trade bill, HR 3, contains provisions that would discourage foreign investment in the US and an amendment, the Gephardt Amendment, that would require automatic retaliation against some trading partners.

The committee urged members to support HR 2150, a substitute bill. "It incorporates the best of HR 3," Cheney said, "but deletes those sections of the bill that are likely to reduce international trade or harm US economic interests."

According to Cheney, the substitute bill imposes strict negotiating objectives on the administration, provides expedited remedy to injured domestic industries, and requires the US trade representatives to investigate and take action against unfair trade practices, such as targeting and dumping in third country markets.

Update on NSFnet

The National Science Foundation established NSFnet to connect NSF supercomputing centers nationwide. The NSFnet backbone has been running since October 1986, connecting the National Center for Atmospheric Research in Boulder, Colorado, and the five NSF National Advanced Scientific Computing Centers.

The NSFnet backbone is currently running at 56K bps, with upgrading to 1.544M bps T1 lines expected by the end of this year.

A number of regional networks, funded at least in part by the NSF, are connected or soon to be connected to the NSFnet backbone.

Previously existing networks being connected are the Merit Computer Network in Michigan; CSNET, an international mail network of computer science departments; and ARPANET.

A status report on the national networks appears in the April 1987 issue of "Forefronts," published by the Center for Theory and Simulation in Science and Engineering at Cornell University.

NAM supports "intellectual property" protection act

In a statement submitted April 30 to the Subcommittee on Monopolies and Commercial Law, the National Association of Manufacturers urged lawmakers to support HR 557, the "Intellectual Property Antitrust Protection Act."

"This bill contains some badly needed changes," the NAM said, "which will increase the value of intellectual property (patents and trade secrets) to United States business and industry, both large and small innovators and inventors."

Evaluating the conduct of patent owners under the antitrust laws, NAM said, "Some courts have characterized the patent system as being inherently in conflict with antitrust goals. Therefore, it is our position the judicial approach has discouraged licensing that would permit intellectual property owners to compete effectively."

The NAM said Section 2 of the bill would amend the antitrust laws to instruct courts not to treat intellectual property license arrangements as per se illegal. "By removing the cloud of per se condemnation from intellectual property licensing practices," NAM stated, "the ability of technology owners to exploit their rights efficiently would no longer be undermined."

NAM says the Antitrust Division of the Department of Justice now recognizes that licensing complements rather than conflicts with the purpose of the antitrust laws, but some court decisions continue to reflect hostility toward intellectual property.

As NAM sees it, the bill merely obliges the court to hear evidence on the actual economic effects of the licensing arrangement to arrive at a finding of unlawfulness using a "rule of reason" analysis.

Fulbright Scholar reminder

The Council for International Exchange of Scholars reminds faculty and scholars that the competition for the 1988-89 Fulbright grants is in progress.

The basic eligibility requirements are US citizenship; PhD or comparable professional qualifications; university or college teaching experience; and, for selected assignments, proficiency in a foreign language.

For applications and deadlines, contact the Council for International Exchange of Scholars, Eleven Dupont Circle NW, Washington, DC 20036-1257; (202) 939-5401.

The Council also invites institutions to submit proposals for visiting scholars through the Fulbright Scholar-in-Residence Program.

A Fulbright Scholar-in-Residence may teach regular courses from a foreign area perspective, serve as a resource person, assist in developing new courses, or participate in special seminars.

The program provides roundtrip travel for the grantee and, for full-year awards, one accompanying dependent; a monthly maintenance allowance; and incidental allowances for travel, books, and services essential to the assignment. The host institution is expected to share some costs.

Detailed program guidelines and proposal forms are available as of June from the address above. The deadline for receipt of proposals is November 1, 1987.

Federal R&D budget increases

Almost all of the 16 percent increase in federal research and development funds proposed in President Reagan's fiscal year 1988 budget fall into the four categories of defense, health, space research and technology, and general science, according to a National Science Foundation Report.

These four categories account for 92 percent of the $67.6 billion allocated for R&D in the 1988 budget.

To obtain a copy of the report "Federal R&D Funding by Budget Function, Fiscal Years 1986-1988," contact Margaret Meredith, Government Studies Group, Div. of Science Resources Studies, NSF, 1800 G St. NW, Washington, DC 20550; (202) 634-4636.

Computer Society NEWS

Editor: Sallie Sheppard, Dept. of Computer Science, Texas A&M University, College Station, TX 77843; (409) 845-5466

Software Engineering TC urges support for Wang Institute

In a unanimous motion passed at its March 31 meeting, the Executive Committee of the Computer Society's Technical Committee on Software Engineering called on industry and governmental agencies to provide alternative funding to maintain the program offered by the Wang Institute of Graduate Studies. Efforts to broaden support for the Tyngsboro, Massachusetts, institute — especially in the New England area — will be coordinated by Susan Gerhart of the Microelectronics and Computer Technology Corporation (MCC), Austin, Texas.

The committee's action was taken in response to the late-February announcement by An Wang that he could no longer fund the institute he founded in 1979. Under the terms of a merger arranged with Boston University, the institute's flagship degree, the master of software engineering, will be discontinued. Traditional business-related part-time graduate programs will be offered by the institute as a branch of Boston University.

The Wang Institute, which admitted its first students in January 1981, will award its last MSE degree in August 1987. During its operation, the institute established a professionally oriented degree program in software engineering targeted to experienced programmers. The curriculum consisted of six core courses, three electives, and two team projects. The content was equally distributed over mathematics, methods, and management topics.

A total of 168 degree students have attended the institute, with 120 expected to earn the MSE before the institute's announced closure. The present faculty has resigned effective August 31. After that time, the physical facilities will be used by Boston University for course offerings in the Merrimack Valley.

In the view of the Software Engineering TC, discontinuing the institute's MSE degree is a great loss to the fledgling software engineering profession. Not only have its graduates taken their places in the profession, but its curriculum and its research and technical support activities have become a worldwide model for similar programs. The institute also served a broader community with its Distinguished Lecture Series and Summer Institute in Computer Science.

The committee believes that this unique institution is too valuable to lose. Failure of the Wang Institute governors and of the software industry to expand support for the MSE continuation will lead to fewer trained software engineers and to dispersion and subsequent loss of expertise in teaching this difficult blend of management and technical subjects.

Board of Governors approves slate of candidates for fall election

At its May 8 meeting in Miami, Florida, the Board of Governors approved the following slate of candidates for three Computer Society offices and 10 board positions to be filled by membership vote this fall.

1988 president-elect (1989 president)
Kenneth R. Anderson
Michael C. Mulder

1988 first vice president
P. Bruce Berra
Helen M. Wood

1988 second vice president
Paul L. Borrill
Joseph E. Urban

1988-89 Board of Governors
Bill D. Carroll
Gerald R. Engel
Lansing Hatfield
Barry Johnson
Duncan H. Lawrie
Raymon P. Oberly
Anthony Pau
David Pessel
Ralph Preiss
V. Thomas Rhyne
Susan L. Rosenbaum
Roland J. Saam
Sallie Sheppard
Bruce Shriver
Harold S. Stone
Akihiko Yamada
Marshall Yovits

Additional nominees can be added to the ballot by membership petition (see *Computer*, January 1981, p. 108, for Computer Society bylaws requirements for petition candidates). Petition candidates must submit their petition signatures (250 voting members for officer nominees, 50 voting members for Board of Governors nominees) and their biographical data, photographs, and position statements to the Computer Society secretary at the following address on or before June 22, 1987:

Secretary
Computer Society Elections
Computer Society of the IEEE
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

Candidates' statements and biographical data will be published in the August 1987 issue of *Computer* and will be included in the August 3, 1987, IEEE ballot mailing. Length limitations for these materials are as follows:

Candidates' statements	
President-elect	350 words
Vice president	250 words
Board of Governors	150 words
Biographical data	
All candidates	200 words

Biographical sketches should cover the following topics in the sequence listed:

- Computer Society activities
- Other professional activities
- Current employment, professional experience, and accomplishments
- Degree(s) and major(s)
- Awards and honors

Nominees should also submit a black-and-white passport-type photograph of themselves for publication along with their statements and biographies.

Committee on Public Policy finds E-mail security pamphlet wanting

David Snow, COPP Subcommittee on Security and Privacy

The E-Mail Association's Infosec pamphlet was reviewed, but not endorsed, by the Computer Society's Committee on Public Policy at its February 23 meeting in San Francisco. According to the committee, the information security guidelines in the pamphlet titled "Electronic Mail and Information Security: An Issue for your Organization?" are incomplete and, in some cases, conducive to insecure practices.

The committee recognizes the need for security in electronic mail, and applauded the Washington, DC, based organization of electronic mail providers for addressing the issue. But the consensus was that the committee should not endorse a weak and incomplete approach to the problem.

The E-Mail Association pamphlet points out the need for organizations using electronic mail to be aware of information security requirements and includes a six-step checklist. It also identifies four special considerations directly affecting the security of electronic mail.

Items on the checklist include

(1) Recognize information as an organizational asset.

(2) Assign the resources to create and implement an information security policy, one that includes electronic mail.

(3) [Implement data] classification and control.

(4) Consider an ownership program [of the classified data].

(5) Establish an education and [security] awareness program.

(6) Ensure [security] measurement and reviews — an ongoing audit.

COPP had no complaint with any of these items but found the list incomplete. For example, item 3 ignores the need for access control and data integrity. Moreover, the issues of peer and source authentication are left out. Committee members also pointed out that the checklist places the entire burden of security on the users of E-mail and leaves out the responsibilities of the developers and providers of the service. The committee believes that the pamphlet should be accompanied by a set of recommended security features to be incorporated into E-mail products and services.

The four special considerations cited in the pamphlet were also found to be faulty. In particular, the password recommendations lead to a false sense of security. Instead of recommending only that the users change their original "welcome" password, the pamphlet should have recommended frequent changes of passwords, say monthly. The pamphlet also instructs users not to use "easy to guess" passwords, but ignores the more sensible approach of having the system assign "random" passwords to users at frequent intervals. Additional considerations that should have been stressed are access control and source and peer authentification.

The committee believes that much of the technology necessary for secure E-mail is available today. What is actually required is a focusing of available technology by the definition of a comprehensive and coherent E-mail security policy. While not endorsing the E-Mail Association's specific recommendations, COPP reaffirmed its desire to cooperate with the association and similar industrial and professional organizations in defining appropriate security policies.

Information on the COPP and its subcommittees is available by circling number 195 on the Reader Service Card at the back of the magazine or by contacting Paul Davis, Secretary, Computer Society Committee on Public Policy, 41 W. Huron St., Jackson, OH 45640; (614) 289-2331 x5794.

Technical Committee on Software Engineering activities outlined

Lorraine M. Duvall, SE-TC Chair

Activities of the Computer Society's Technical Committee on Software Engineering include (1) organizing workshops and conferences, (2) developing the Fundamental Concepts on Software Engineering article series for publication in *IEEE Software*, (3) sponsoring standards activities, (4) sponsoring best paper awards, and (5) publishing a TC newsletter.

Conferences and workshops. The divergence of opinions expressed at the Ninth International Conference on Software Engineering, held March 31-April 2 in Monterey, California, is typical of the software engineering field (see report in Conferences Department). Practitioners and researchers all seek the production of reliable software, produced on-time for a reasonable cost, but do not always agree on mechanisms for achieving this goal.

In addition to its ongoing involvement in the Software Engineering Conference, the committee sponsors other technical meetings. It recently initiated an annual workshop dedicated to focused topic areas. This year's workshop will be on reusability and will be held July 28-31 at the Minnowbrook Conference Center, Blue Mountain Lake, New York. It will be coordinated with the annual Minnowbrook Workshop on Software Performance Evaluation, which has been organized by Amrit Goel of Syracuse University for the past nine years.

Article series. One major project tackled by the TC on SE is development of the Fundamental Concepts in Software Engineering series for *IEEE Software*. The project grew out of a September 1982 meeting of software-engineering leaders to provide recommendations for advancing the field. The participants had become concerned about rapid changes in the environment in which software is developed and the ability of software engineering to deal with the changes effectively. John D. Musa's report of this meeting, "Software Engineering: The Future of a Profession," was published in the January 1985 *IEEE Software*.

In the area of software engineering technology creation, the report's highest priority recommendation was to commission a "best idea" article series. Each article would explore a best idea, as adjudged by a panel of experts, from the standpoint of how it was conceived, how it has matured over the years, and how it has been applied.

An editorial board was created to implement the series. The current board consists of Editor-in-Chief Norman Schneidewind of the Naval Postgraduate School, Bruce Barnes of the National Science Foundation, Robert L. Glass of the Software Engineering Institute, Elaine J. Weyuker of New York University, and Marvin Zelkowitz of the University of Maryland.

The first article in the series, "Structured Programming: Retrospect and Prospect" by Harlan D. Mills of IBM, was published in *IEEE Software* in

November 1986. Additional articles, planned for publication in the near future, will cover software engineering processes, the theory and practice of software testing, structured analysis and design, and microprogramming.

Standards. The Software Engineering Standards Subcommittee is very active in developing software engineering guidelines and standards. The subcommittee, chaired by John W. Horch of Teledyne Brown Engineering, is currently sponsoring standards activities in the areas of measures for producing reliable software, the use of Ada as a program design language, software verification and validation plans, software design descriptions, software reviews and audits, configuration management, software productivity and quality metrics, project management plans, software maintenance, software certification, and software user documentation.

Coordination and information. The technical committee coordinates activities and holds joint meetings with SIGSoft, the ACM Special Interest Group on Software Engineering. It also maintains a close liaison with the Computer Society's software-related publications.

For additional information on the Technical Committee on Software Engineering, circle number 194 on the Reader Service Card at the back of the magazine or contact Lorraine M. Duvall, Duvall Computer Technology, 1317 N. Madison St., PO Box 568, Rome, NY 13440; (315) 337-1564.

Nominations sought for McDowell Award

The chair of the W. Wallace McDowell Awards Committee has requested nominations for the 1987 award. Established in 1976 through an IBM grant, the McDowell Award is the Computer Society's highest technical achievement award.

Nominations must include the name and address of the nominee, the names and addresses of three persons familiar with the nominee's work, a proposed citation, and at least five paper, book, or patent references to the work for which the nominee is to be recognized.

Send nominations by September 1 to Rex Rice, Chair, W. Wallace Award, Rice Consulting, 1011 Sierra Drive, Menlo Park, CA 94025.

Auditor's report: Computer Society 1986, 1985 financial statements

Continuing a policy initiated three years ago, Coopers & Lybrand has examined the Computer Society's financial records. The accounting firm's report for the years ended December 31, 1986 and 1985, appears below.

Board of Governors of
The Computer Society of the IEEE

We have examined the balance sheets of the Computer Society of the IEEE as of December 31, 1986 and 1985, and the related statements of revenue, expenses,

The Computer Society of the IEEE
Balance Sheets
December 31, 1986 and 1985

Assets	1986	1985
Current assets:		
Cash, including interest-bearing accounts	$ 558,200	$ 304,900
Investments (Note 3)	1,502,100	1,811,600
Accounts receivable, less allowance of $111,700 in 1986 and $81,000 in 1985	670,300	581,800
Receivable from Institute of Electrical and Electronics Engineers, Inc. (Note 7)	71,300	84,000
Conference receivables	423,100	28,600
Conference advances	139,000	188,900
Inventory	420,000	316,800
Prepaid expenses and other assets	697,600	557,600
Total current assets	4,481,600	3,874,200
Fixed assets, net (Note 4)	3,393,200	3,497,800
Total Assets	$7,874,800	$7,372,000
Liabilities and Fund Balance		
Current liabilities:		
Accounts payable and accrued expenses	$1,074,800	$ 806,500
Notes payable (Note 5)	33,000	29,900
Deferred income:		
Membership fees and subscriptions	2,521,500	2,323,700
Conferences	127,600	104,800
Advertising and other	118,500	181,500
Total current liabilities	3,875,400	3,446,400
Notes payable, less current portion	1,643,100	1,675,100
Total liabilities	5,518,500	5,121,500
Fund Balance:		
Undesignated	2,277,000	2,181,600
Designated for technical committees (Note 9)	79,300	68,900
Total fund balance	2,356,300	2,250,500
Total liabilities and fund balance	$7,874,800	$7,372,000

and changes in fund balance for the years then ended. Except as explained in the following paragraph, our examinations were made in accordance with generally accepted auditing standards and, accordingly, included such tests of the accounting records and such other auditing procedures as we considered necessary in the circumstances.

As discussed in Note 8 to the financial statements, the society reports revenue and expenses of all conferences partially or entirely sponsored by the society. However, for certain conferences partially sponsored by the society, revenue and expenses are reported solely on the final representations of the external conference administrators. Consequently, the society does not maintain or control supporting information for all conference revenue and expenses. Therefore, we have not been able to examine sufficient documentary evidence with respect thereto.

In our opinion, except for the effects of such adjustments, if any, as might have been determined to be necessary had we been able to examine sufficient documentary evidence with respect to conference revenue and expenses, the financial statements referred to above present fairly the financial position of the Computer Society of the IEEE as of December 31, 1986 and 1985, and the results of its operations for the years then ended, in conformity with generally accepted accounting principles applied on a consistent basis.

Coopers & Lybrand

1800 M Street, NW
Washington, DC 20036
February 27, 1987

The Computer Society of the IEEE
Statements of Revenue, Expenses, and Changes in Fund Balance
for the years ended December 31, 1986 and 1985

	1986	1985
Revenue:		
Computer Society membership fees	$ 720,500	$ 701,400
Periodical subscriptions and other publication activities	7,629,400	6,946,400
Conventions, conferences, and other technical activities	3,951,300	2,739,200
National Computer Conference distribution (Note 7)	140,400	484,600
Allocated transfers (Note 7)	—	95,400
Interest income	71,400	151,300
Other income	262,000	160,100
Total revenue	12,775,000	11,278,400
Expenses:		
Periodical and publication activities	7,467,900	7,472,000
Conventions, conferences, and other technical activities	3,710,200	3,123,400
Administration	1,491,100	1,196,700
Other expenses	—	16,600
Total expenses	12,669,200	11,808,700
Excess (deficit) of revenue over expenses	105,800	(530,300)
Fund balance at beginning of year	2,250,500	2,780,800
Fund balance at end of year	$ 2,356,300	$ 2,250,500

Notes to financial statments

1. Organization and purpose

The Computer Society of the IEEE is organized within the Institute of Electrical and Electronics Engineers, Inc., an organization exempt from income tax pursuant to Internal Revenue Code Section 501(c) (6). Within the bylaws of IEEE, delegation of the responsibility for the society's operations has been placed with the society's Board of Governors and Executive Committee. The society's constitution states that "The Society shall be scientific, literary, and educational in character. The Society shall strive to advance the theory, practice, and application of computer and information processing science and technology, and shall maintain a high professional standing among its members. The Society shall promote cooperation and exchange of technical information among its members and to this end shall hold meetings for the presentation and discussion of technical papers, shall publish technical journals, and shall, through its organization and other appropriate means, provide for the needs of its members."

2. Summary of significant accounting policies

A. *Reporting entity*

The accompanying financial statements include all society accounts maintained at the society's offices in Washington, DC, Los Alamitos, California, Brussels, Belgium, and certain accounts maintained at IEEE Headquarters. The society has chapters which operate directly under IEEE and, therefore, the accompanying financial statements exclude the accounts of these chapters.

B. *Income recognition*

Income from annual membership fees and periodical subscriptions is recognized during the year to which it pertains. Revenues and expenses of conferences are generally recognized in the year in which the conference is held.

C. *Fixed assets and depreciaiton*

Fixed assets are recorded at cost when purchased. The society provides for depreciation of fixed assets by charges to revenue at rates considered adequate to amortize the cost of such assets over their estimated useful lives (5 to 10 years for office furniture and equipment; 35 years for buildings) on a straight-line basis.

When fixed assets are retired or otherwise disposed of, the property and accumulated depreciation accounts are relieved of the applicable amounts and any profit or loss is reflected in revenue.

D. *Inventory*

Inventory consists of tutorial books published by the society and is stated at the lower of cost or market. Cost is determined on an average cost basis.

3. Investments

Investments consist of unrestricted deposits with IEEE and bear interest based on United States Treasury Bill rates and are carried at cost which approximates market.

4. Fixed assets

Fixed assets as of December 31 are shown in Table 1.

Depreciation expense amounted to $226,600 and $183,900 in 1986 and 1985, respectively.

5. Notes payable

Notes payable as of December 31 were as shown in Table 2.

The note payable due May 1, 1990, is collateralized by a first lien on all gross revenues of the society and a mortgage on the Washington, DC, property. Repayment is in graduating amounts through the term of the note with the balance payable on May 1, 1990.

The note payable due May 24, 1988, is collateralized by a deed of trust on one of the two buildings located in Los Alamitos, California. Repayment is in equal monthly payments of $5,200 with the balance payble on May 24, 1988.

Interest expense amounted to $165,500 and $125,500 in 1986 and 1985, respectively.

Annual maturities of long-term debt are as follows: 1987—$33,000; 1988—$485,100; 1989—$31,000; and 1990—$1,127,000.

6. Pension plan

The society is a member of a defined-benefit pension plan sponsored by IEEE. The IEEE plan covers substantially all IEEE employees, including those of the society. It is the policy of IEEE to fund pension costs accrued.

Statement of Financial Accounting Standards No. 36, "Disclosure of Pension Information," requires that certain disclosures be made of the actuarial present value on accumulated plan benefits and the fair value of net assets available for plan benefits. Such disclosure is not presented for the society because the structure of the IEEE plan does not readily permit the plan's assets and accumulated benefits data to be determined for each individual society. Based upon actuarial valuations of the IEEE plan, utilizing assumed rates of return of 8¾% and 8½% for 1986 and 1985, respectively, the fair value of net assets available for plan benefits exceeded the actuarial present value of vested and nonvested accumulated plan benefits at January 1, 1986 and 1985.

The society was allocated approximately $38,000 and $80,720 for pension expense in 1986 and 1985, based upon its respective share of total payroll and benefit costs.

7. Related-party transactions

A. *Institute of Electrical and Electronics Engineers, Inc.*

Certain general and administrative expenses incurred by IEEE Headquarters and charged to the society amounted to $624,000 in 1986 and $567,700 in 1985. Other transactions undertaken in the normal course of business between the society and IEEE have been reflected in the society's financial statements.

B. *American Federation of Information Processing Societies (AFIPS)*

The society is one of eleven members of AFIPS and is represented by three members on the AFIPS Board of Directors. Any surplus which AFIPS generates (primarily from the National Computer Conference) is allocated to the sponsoring societies based upon their respective board representation. The IEEE Foundation was allocated $95,400 in 1985 of the surplus generated by AFIPS. The IEEE Board of Directors authorized a transfer of the $95,400 to the society in 1985.

8. Conferences sponsored by the society

The society reports its share of revenues and expenses on all conferences partially or entirely sponsored by the society. Supporting documentation of the financial activity of those conferences entirely sponsored by the society is maintained at the society's offices. However, for certain conferences sponsored by the society, supporting documentation is maintained by the conference administrators and only a summary final report of the financial activity is received by the society.

9. Designated fund balance

The Board of Governors has designated a portion of surplus funds received from recurring conferences for use by technical committees in accordance with the society's policy on meeting surplus accounts. The designated amounts are calculated based upon a formula contained in the policy, but in no case can they exceed $15,000 per technical committee conference.

Table 1. Fixed assets as of December 31.

	1986	1985
Land	$1,334,400	$1,334,400
Building and improvements	1,800,800	1,795,800
Warehouse equipment	25,100	25,100
Office furniture and equipment	1,055,900	938,900
	4,216,200	4,093,300
Accumulated depreciation	(823,000)	(596,400)
	$3,393,200	$3,497,800

Table 2. Notes payable as of December 31.

	Annual interest rate	1986	1985
Note payable, due March 1, 1990	Prime 12.0%	$1,212,200	$1,235,500
Note payable, due May 24, 1988		463,900	469,500
		1,676,100	1,705,100
Less: amount due within one year		33,000	29,900
		$1,643,100	$1,675,100

HARDWARE ACCELERATOR
386 TURBO
INTELLECTUAL ACCELERATOR
COMPUTER
Super Machines
Sprinters or Stayers?
Computer Graphics and applications
MICRO
MULTIPROCESSING: LOW-COST, FAST TASK HANDLING
Design & Test
VHDL: The VHSIC Hardware Description Language
EXPERT
Software
REV.8 SN.
The Computer Society
of the IEEE

Upgrade your knowledgeware.

Join the Computer Society today.

Where can you find, from one source, the most complete, current, reliable information on computer technology today? The Computer Society of the IEEE.

Our periodicals, tutorials, proceedings, and standards are your keys to knowledge in any area of computer technology.

The Computer Society is dedicated to the dissemination and support of advanced technologies and to the professionals who create them. It is your most important personal resource for keeping current in today's complex world of computers and information systems.

Don't hesitate any further. Join us today.

member benefits:

- **Automatic subscription to *COMPUTER* magazine**
  This authoritative yet readable monthly brings you industr surveys, tutorials, and application perspectives across the breadth of the computer field.
- **Discounts on Computer Society periodicals and othe IEEE publications**
  Subscribe at reduced rates to the eight other specialized Computer Society transactions and magazines and many IEEE publications.
- **Electronic mail**
  Subscribe at low member rates to COMPMAIL+, an international electronic mail service that includes major news databases, bulletin boards, and gateways to other electronic networks.

Computer
- hardware and software design
- tutorials
- surveys
- product and industry news (monthly)

Automatic with membership

IEEE Computer Graphics and Applications
- displays
- computer-aided design and manufacturing
- computer animation
- solid modeling
- hardware/software tools for rendering and graphics

(monthly)

Member rate: $26/year

IEEE Design&Test of Computers
Cost-Effective Chip Diagnostics

IEEE Design & Test of Computers
- design automation
- computer-aided design
- workstations and test equipment
- design for testability
- test strategies for memory and logic
- test of components, assemblie and systems
- quality assurance
- VLSI

(bimonthly)

Member rate: $17/year

Discounts on Computer Society books

The Computer Society Press has more than 650 titles in print, including tutorial texts, conference proceedings, and hardware/software standards. Members receive discounts up to 50%—a savings that often pays for the price of membership on the first book order.

Discounts on Computer Society conference registration

To bring together professionals and users from the spectrum of disciplines, we annually sponsor more than 85 conferences worldwide on topics ranging from artificial ntelligence to zero-defect design. These meetings range rom small, specialized workshops with limited attendance o large industry-oriented conferences replete with exhibits, such as the Design Automation Conference or the nternational Test Conference. But whether small or large, he guiding principle is always high technical quality.

- **Technical committees and standards task forces**
  Your participation in the activities of the more than 30 technical committees provides you with a technical forum and an opportunity for stimulating interactions with colleagues from around the world who work in your specialty.
- **Local chapter involvement**
  Your prime source for continuing education, idea exchange, and professional contacts is the local Computer Society chapter. We support over 100 such chapters worldwide.
- **Opportunities for contributing your own talents**
  The authority of our programs is based on a high level of member involvement: society members edit publications, organize conferences, present papers, write and review articles, work on curricula or accreditation matters, develop standards, and oversee the conduct of the society's business.

IEEE Expert

- expert and intelligent systems
- artificial intelligence
- database and data engineering
- medical and industrial applications
- natural language processing
- planning and problem solving
- automated reasoning
- knowledge engineering

(quarterly)

Member rate: $12/year

IEEE Micro

- microprocessor and board-level design
- system integration
- hardware and software environments
- VLSI technology
- reviews of commercial microprocessor, memory, and logic chips
- standards and computer law

(bimonthly)

Member rate: $17/year

IEEE Software

- programming environments
- compilers and languages
- software engineering and programming methodologies
- distributed, network, and parallel systems
- algorithms for multiprocessor computers
- data engineering and database software
- debugging and testing

(bimonthly)

Member rate: $16/year

NOW!

Special offer to members only

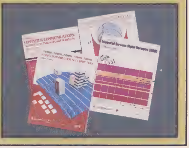
To new and current members, we are offering two special collections of Computer Society Press books at less than regular member prices.

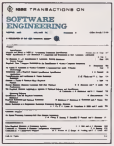
Software Development Collection

Networking and Communications Collection

Each of these special collections consists of four of our most popular titles in two of the fastest-growing areas of computer technology.

Use order form on back page. Note: to take advantage of this offer, you must either be a member now or submit your application with the order.





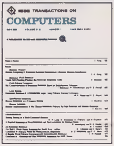
TRANSACTIONS ON SOFTWARE ENGINEERING

IEEE Transactions on Software Engineering

- specification, design, development, management, testing, maintenance, and documentation of software systems
- programming methodology
- programming environments
- software project management
- hardware and software monitoring
- programming tools

(monthly)

Member rate: $18/year



TRANSACTIONS ON COMPUTERS

IEEE Transactions on Computers

- theory, design, and practice of computer systems
- circuits, systems, and networks
- VLSI and digital devices
- fault-tolerant computing
- testing and performance evaluation
- computational methods
- computer applications

(monthly)

Member rate: $18/year

TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

IEEE Transactions on Pattern Analysis and Machine Intelligence

- knowledge representation
- inference systems
- statistical and structural pattern recognition
- group searching
- robotics
- natural language processing
- image processing

(bimonthly)

Member rate: $15/year

THE COMPUTER SOCIETY OF THE IEEE

COMPUTER SOCIETY
Membership Application

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Yes, I want to take advantage of the special book offer available only to members.

☐ I have completed the membership application.
☐ I am already a member. My membership number is: ____________

☐ SOFTWARE DEVELOPMENT COLLECTION—
The Outstanding Series for Software Engineers

New Paradigms in Software Development
by William Agresti

Human Factors in Software Development
by Bill Curtis

JSP and JSD–The Jackson Approach to Software Development
by John Cameron

Software Development Environments
by A. Wasserman

Order number PSD-11
Non-member price: $175.00
Regular member price: $105.00
Special offer: $90.00
Handling charge: $6.00 additional. Foreign orders add 15%.

☐ NETWORKING AND COMMUNICATIONS COLLECTION—Four Best Sellers by William Stallings

Computer Communications–Architectures, Protocols, and Standards

Local Network Technology

Reduced Instruction Set Computers

Integrated Services Digital Networks

Order number: PNC-12
Non-member price: $158.00
Regular member price: $107.00
Special offer: $92.00
Handling charge $6.00 additional. Foreign orders add 15%.
Add remittance to membership fees.

Schedule of Fees

Membership dues and publication subscriptions are annualized to December 31. Depending upon the date your application is received by the Computer Society, pay the full annual amount or one-half the annual amount as indicated below. Membership expires December 31.

	Half-Year Mar 1-Aug 31	Full-Year Sept 1-Feb 28
1 I don't belong to the IEEE and I want to join just the Computer Society	☐ $19.50	☐ $39.00
2 I already belong to the IEEE and I want to join the Computer Society. IEEE Member Number ____________	☐ $ 7.50	☐ $15.00

3 I don't belong to the IEEE and I want to join both the Computer Society and the IEEE*
(Total amount to be paid includes annual dues and regional assessment, if any)

I reside in Region 1-6 (United States)	☐ $41.00	☐ $82.00
I reside in Region 7 (Canada)	☐ $38 50	☐ $77.00
I reside in Region 8 (Europe, Africa, or the Middle East)	☐ $37.00	☐ $74.00
I reside in Region 9 (Latin America)	☐ $33.50	☐ $67.00
I reside in Region 10 (Asia and Pacific)	☐ $34.00	☐ $68.00

*ACM members who join both IEEE and the Computer Society are entitled to a deduction of $5 off the full-year rate, $2 50 off the half-year rate

4 In addition to my automatic subscription to *COMPUTER*, I would also like to subscribe to:

	(issues per year)		
IEEE Computer Graphics and Applications (3061)	12	☐ $13 00	☐ $26.00
IEEE Design and Test (3111)	6	☐ $ 8.50	☐ $17.00
IEEE Expert (3151)	4	☐ $ 6.00	☐ $12.00
IEEE Micro (3071)	6	☐ $ 8.50	☐ $17.00
IEEE Software (3121)	6	☐ $ 8.00	☐ $16.00
Transactions on Computers (1161)	12	☐ $ 9.00	☐ $18 00
Transactions on Pattern Analysis and Machine Intelligence (1351)	6	☐ $ 7.50	☐ $15 00
Transactions on Software Engineering (1171)	12	☐ $ 9.00	☐ $18.00
Journal on Lightwave Technology (4301)	12	☐ $ 9 00	☐ $18.00
Journal of Robotics and Automation (4401)	4	☐ $ 5.00	☐ $10.00
Journal of Solid State Circuits (4101)	4	☐ $ 4 00	☐ $ 8.00
Proceedings of the IEEE (5011) (available only to IEEE members)	12	☐ $ 8.50	☐ $17.00

Total amount remitted with this application $____________ (remit in U.S. dollars only)

☐ Check or money order enclosed. Make payable to IEEE.
Mail to Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720.

☐ VISA ☐ MasterCard ☐ American Express

Charge Card Number ____________ Exp Date Mo | Yr

I hereby make application for Computer Society membership and, if elected, will be governed by IEEE's and the Society's Constitutions, Bylaws, and Statements of Policies and Procedures.

Full signature ____________ Date ____________

MAILING ADDRESS

First Name ____________ Middle initial(s) ____________ Last name ____________

Street address ____________ City ____________ State/Country ____________ Zip ____________

EDUCATION (highest level completed) Course ____________ Degrees received ____________ Date ____________

Name of educational institution ____________ Date of birth ____________ ☐ Male ☐ Female

OCCUPATION ____________ Title or position

Firm name ____________

Firm address ____________

City ____________ State/Country ____________ Zip ____________

Affiliate society membership no. (if applicable) ____________ Grade ____________

REFERENCE (an IEEE member; if unknown, a managerial person who knows you professionally)

Name (print in full) ____________ IEEE Member No (if applicable) ____________

Position ____________

Street address ____________

City ____________ State/Country ____________ Zip ____________

MAIL TO: COMPUTER SOCIETY, 10662 LOS VAQUEROS CIRCLE, LOS ALAMITOS, CA 90720–2578

C187

Editor: Edmund L. Gallizzi, Computer Science Dept., Eckerd College, St. Petersburg, FL 33733, (813) 867-1166 x272

The Computer Society's conference program

James H. Aylor, Vice President for Conferences and Tutorials

The Computer Society's conference program is an elaborate and delicate interaction of many individuals, technical interest groups (called technical committees within the society), and other professional societies both at home and abroad.

The Computer Society sponsored or cosponsored over 50 technical meetings and cooperated in an additional 30 technical meetings in 1986. Sponsoring implies not only a significant component of technical involvement but also a significant degree of financial support. Cooperation implies strong technical support for the meeting through the expertise of the members of the society but with no financial involvement.

The approval of all of the technical meetings of the society is handled through the vice president for conferences and tutorials and the Conferences and Tutorials Board. This structure establishes a focal point for the interaction between technical meetings and the technical community responsible for creating the meeting and is essential to the control of a $4 million conference operation.

Three forms. The society's technical meetings have three forms — the conference, symposium, and workshop.

Conferences, which are meetings with more than 300 attendees, include the very successful Design Automation Conference (DAC), International Test Conference, International Conference on Software Engineering, and International Conference on Artificial Intelligence Applications (CAIA). DAC is the largest with an attendance of 3400 and has the world's largest exhibit in the area of design automation. CAIA, initiated in 1985, is the newest and attracts 600 attendees.

The workshop is a forum for the exchange of information related to a very focused technical area. Its attendance is typically less than 100, and it often has no published proceedings. This format encourages the exchange of key technical information among recognized individuals in a field that would not be possible if the discussion were to be published.

Symposia are meetings with an expected attendance of 100-300.

The Computer Society Press publishes the proceedings for the majority of the sponsored conferences to ensure the quality and appearance of the document.

International aspects. Although the society has always had a strong international component in all of its activities, much more emphasis has been placed recently on making the technical meeting truly international. Many of our conferences have international members on their committees; a number of our meetings are being rotated through other countries of the world; and new conferences are being established through a cooperative effort between the Computer Society and technical societies abroad.

In 1984, the First International Conference on Computers and Applications was established with the China Computer Society and held in China. It is being repeated this June. CompEuro, which is being held in Hamburg this May, was established as a joint venture between Region 8 of the IEEE and several European societies to stimulate cooperation among computer professionals in Europe. The Computer Society will host the biannual conference of the International Association of Pattern Recognition in 1990 in Atlantic City, New Jersey.

Directions. The strength of the successful conference program is in the quality and dedication of the technical community represented by the members of the society, in particular the members of the various technical committees. The overall goal of the Conferences and Tutorial Board is to provide the most effective structure and services to support these members. There is presently a board committee established to assess current activities within the conferences area and to establish the future direction for the area. Its report is due in November.

The conference program is a strong and growing activity within the society. Involvement in conference planning and operation is exciting and technically rewarding. The society is always looking for new volunteers and encourages you to join in conference activities.

Conferences—a new magazine department

The goal of the Conferences department, as well as the accompanying Calendar and Call for Papers sections, is to support the conference system by

- presenting the conference calendar along with articles describing special events in upcoming conferences;
- reporting important results of conferences; and
- maintaining a continuing dialog concerning the structure, importance, operation, etc., of the conference system.

The upcoming conference calendar and articles will provide a conduit from the conference committees to prospective conference attendees. The articles will provide balanced coverage for both major conferences in which the Computer Society has a large commitment and smaller conferences that have particularly important and innovative agendas. The post-conference reports will focus on particularly important, timely, and controversial results, events, quotes, awards, etc., from conferences that might not be part of the conference proceedings. To implement the pre-conference articles and the post-conference reports, each of the conference committees for Computer Society-supported conferences will be contacted to request reporting assistance. Since neither the magazine staff nor myself can attend all the conferences, a small cadre of volunteer reporters will be developed. If you would like to be part of that group, or would like to submit a short conference report or suggestions, please contact me.

From time to time, special articles concerning the conference system will be presented. The first of these, by Jim Aylor, the Computer Society's vice president for conferences and tutorials and chairperson of the Conferences and Tutorials Board, follows.

—Edmund Gallizzi

Fly-by-wire airliners a topic of FTCS-17

Technology is on the threshold of providing large automated systems that will operate in life-critical environments and will impact major portions of society. One such system is the next generation of airliners, which includes the 7J7, that will allow fully automated landing systems as well as systems to automatically taxi to any gate at major airports. These airplanes will be an integrating topic across the panels at the 17th Annual International Symposium on Fault-Tolerant Computing, to be held July 6-8 in Pittsburgh, Pennsylvania.

The promise of large automated systems will be the focus of a panel, "Design of Large Automated Systems," and of a keynote address by Raj Reddy, director of the Robotics Institute at Carnegie Mellon University, on robotics and intelligent systems. The requirements of ultra-reliability are the topic of another panel, "Reliability Modeling of Life-Critical Systems," and of the second keynote address by Larry Druffel, director of the new Software Engineering Institute, on reliability issues in software engineering.

The members of the panels tentatively include representatives from the computer and aerospace industries, NASA, FAA, and universities. Additionally, 48 papers on fault-tolerant topics will be presented.

A limited number of travel scholarships supported by industry will be provided for graduate students. For scholarship and conference information, contact John Shen, Carnegie Mellon University, Dept. of Electrical and Computer Engineering, Pittsburgh, PA 15213; (412) 268-3601.

Report on the Ninth Annual International Conference on Software Engineering

Lorraine M. Duvall, Duvall Computer Technologies
David Preston, IIT Research Institute

Over 1200 attendees took part in the stimulating and, at times, provocative Ninth Annual International Conference on Software Engineering. The conference, held March 30-April 2 in Monterey, California, also attracted 600 participants for five tutorials and 45 exhibitors for a tools fair.

Theme and structure. The theme of the conference, "Formalizing and Automating the Software Process," was chosen to highlight the importance of focusing on the software creation and evolution processes. Additionally, the conference organizers actively pursued participation of the database and artificial intelligence communities because of the increasing effect of these disciplines on software engineering.

The conference was structured in three concurrent tracks: process, formalism, and automation. Each track included technical presentations, focused panels, and tools demonstrations. The panels raised the visibility of work performed in the smaller workshops and provided a perspective by the leaders in the field.

Best Paper Awards were presented by Kouichi Kishida, program co-chair, to Dewayne E. Perry of AT&T Bell Laboratories for "Software Interconnection Models" and to K. Schwan, R. Ramnath, S. Vasudevan, and D. Ogle of Ohio State University for "A System for Parallel Programming."

Opposing views. The technical sessions were introduced by Robert Balzer, program co-chair, who explained the Program Committee's perspective. Constructing software as a commodity and managing it as a product has, he said, solved the missed-budget and schedule problems of the 60's and 70's. (This was disputed later by one attendee.) Now the post-delivery problem and our inability to maintain software systems must be addressed.

The fundamental causes of these problem, Balzer said, are inaccessibility of design information and failure to recognize the iterative nature of the maintenance activity. The answers lie in building a corporate memory and in separating maintenance from performance. The process must be formalized, automated, and moved from a mathematical discipline to a design discipline. A main challenge is to integrate the artificial intelligence, database, and software engineering technologies.

In the first plenary address, entitled "Software Processes are Software Too," Lee Osterweil presented the thesis that there is no fundamental difference between the software process and other processes. He maintained that software engineering research should be directed towards the creation of a process programming language, the construction of a compilation and interpretation system for programs written in it, and the use of these tools in the description of key software processes.

In the second plenary address, Manny Lehman responded with an opposing viewpoint. He expressed strong reservations about process programming and concern over its current emphasis in research. According to Lehman, we need to extract the important and worthwhile elements of process programming. When thoughtfully applied, it can be a useful tool for modeling parts of the process, but it's "too seductive," he said, "to be a subject of graduate student research."

During the question and answer session, Harlan Mills, conference chair for the First International Conference on Software Engineering, said he agreed with Balzer on the state of software engineering. He stated that the social process of the conference was freezing out the intellectual aspect of software engineering. To prove his point, Mills

Hewlett-Packard CEO to keynote DAC 87

John A. Young, Hewlett-Packard president and chief executive officer, will discuss the engineering community's response to today's global economy at the 24th annual Design Automation Conference. His keynote address, "Emerging Imperatives for Engineers," will examine the challenges and expectations created for engineers and managers by the shortening of product life cycles and the shifting nature of competitive positions as technology spreads rapidly across national boundaries.

Young, who joined Hewlett-Packard in 1958 after earning a master's in business administration from Stanford University, has served as the firm's president since 1977 and as its CEO since 1987.

The Design Automation Conference, to be held June 28-July 1 in Miami Beach, Florida, will feature 98 exhibits and 130 technical papers and presentations. Paper topics include IC and PCB layout and fabrication, databases, system-level design aids, silicon compilation, and expert systems. The agenda also includes three tutorial sessions and five panels discussions.

For information on DAC registration, contact Pat Pastilli at MP Associates, 7366 Old Mill Trail, Boulder, CO 80301; (303) 530-4333.

asked how many ACM Turing Award winners were in the audience. No hands were raised in response to his question.

Discussion and reaction. Balzer initiated a brown bag "open mike" session on the final day of the conference. In his opening remarks, he articulated three principal questions the conference addressed: Where is the field? Where is it headed? How do we get there? He also stated three assumptions made by the Conference Committee: The process is important; the process must be explicitly represented; and the process must be facilitated through automation.

A panel on formal specifications brought out the opinion that the development process is completely formalized in the US security community but not in the remainder of the US. This is in contrast to Europe where most software development is partially formal. Don Good indicated that fear growing out of national security issues is the primary motivator for current use of formal methods in the US. Formal methods will not become more widely used until there are strong economic motivations.

Within widely varying views on artificial intelligence, perhaps the most realistic perspective was offered by Barry Boehm. Looking to AI as the philosophical touchstone will continue to raise unrealistic expectations with concomitant disappointments, Boehm said. Yet, if you ignore AI, you will miss a potential solution to some of software engineering's problems.

Future conferences. The 10th International Conference on Software Engineering will be in Singapore April 11-15, 1988. The 11th ICSE will be held in Pittsburgh, Pennsylvania, in May 1989. The Steering Committee for the ICSE is pursuing alternate sites for the 12th and 13th conferences in 1990 and 1991.

To be considered as a conference site, a proposing country must have a major computer society, be able to draw 500-600 attendees from the host country, have adequate conference facilities, and propose leaders in the software field to provide direction for the conference.

Countries and organizations that would like to be considered for the 12th or 13th ICSE should write a letter of intent to Lorraine M. Duvall, 1317 N. Madison St., PO Box 568, Rome, NY 13440, USA.

CASE Developers

Go Ahead
Stay Ahead

At Wang, we're pushing the state-of-the-art in computer science research and development. As a member of our Advanced Workstation Development Group, you'll be part of a growing cadre of technical professionals who are advancing this exciting technology. We offer a hands-on environment with total involvement and the opportunity to see the results of your effort.

Wang is currently persuing the development of Computer Aided Software Engineering tools for systems and applications developers built upon our new workstation technology. We seek talented Software Engineers, with a minimum of 3 years of experience, who are dissatisfied with most currently available software development environments and have the ideas and desire to work on a new, sophisticated software engineering environment and associated tools.

We are looking for individuals with experienced backgrounds in one or more of the following areas:

- syntax directed editing
- incremental compilation techniques
- prototyping tools
- compilers and compiler-based technology
- symbolic debuggers
- software design tools
- configuration and revision management
- object-oriented design and programming
- program databases

If you're an experienced Software Engineer who is seeking a continuing state-of-the-art challenge, we should be talking about a future with Wang, where you'll go ahead to stay ahead. To respond, please send a resume or letter detailing your experience to Jackie Hall, Human Resources Manager, Wang Laboratories, Inc., One Industrial Ave., M/S 014-02A, Lowell, MA 01851.

We are an affirmative action employer.

WANG

People Make It Work.

Moving?

PLEASE NOTIFY
US 4 WEEKS
IN ADVANCE

Name (Please Print)

New Address

City **State/Country** **Zip**

MAIL TO:
IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854

ATTACH
LABEL
HERE

- **This notice of address change will apply to all IEEE publications to which you subscribe.**
- **List new address above.**
- **If you have a question about your subscription, place label here and clip this form to your letter.**

CALENDAR

June 1987

NCC-87, National Computer Conference (Computer Society, AFIPS, ACM, DPMA, SCS), June 15-18, Chicago. Contact AFIPS, 1899 Preston White Dr., Reston, VA 22091; (703) 620-8900.

1987 Holm Intensive Course on Electrical Contacts (IEEE), June 15-19, San Francisco. Contact Kate Scheld, IEEE Headquarters, 345 E. 47th St., New York, NY 10017-2394; (212) 705-7405.

Second Annual Conference on Structure in Complexity Theory (Computer Society, ACM), June 16-19, Ithaca, New York. Contact Alan Selman, College of Computer Science, 161 Cullinane Hall, 360 Huntington Ave., Boston, MA 02215; (617) 437-8688.

1987 IEEE Annual Meeting, June 18-19, New York. Contact Betty J. Stillman, IEEE Corporate Services, tenth floor, 345 E. 47th St., New York, NY 10017; (212) 705-7757 or Adeline T. Zeni, IEEE Headquarters, 345 E. 47th St., New York, NY 10017; (212) 705-7304.

IEEE First Annual International Conference on Neural Networks, June 21-24, San Diego, California. Contact N. Feldman, 3770 Tansy St., San Diego, CA 92121; (619) 453-6222.

1987 Spring Conference of the National Society for Computer Applications in Engineering, Planning, and Architecture (CEPA), June 22-23, Washington, DC. Contact Patricia C. Johnson, CEPA, 15713 Crabbs Branch Way, Rockville, MD 20855; (301) 926-7070.

1987 Device Research Conference (IEEE), June 22-24, Santa Barbara, California. Contact Michael S. Adler, Electron Devices Society, General Electric Co., PO Box 8, Room KW-C328, Schenectady, NY 12301; (518) 387-5882 or Jerry Woodall, IBM T.J. Watson Research, PO Box 218, Yorktown Heights, NY 10598; (914) 945-1568.

Second Annual Symposium on Logic in Computer Science (Computer Society, ASL, EATCS), June 22-25, Ithaca, New York. Contact Ashok K. Chandra, IBM T.J. Watson Research Center, PO Box 218, Yorktown Heights, NY 10598; (914) 945-1752.

ICIASF-12, 12th International Conference on Instrumentation in Aerospace Simulation Facilities (IEEE, NASA), June 22-25, Williamsburg, Virginia. Contact William W. Hunter, Jr., NASA, Langley Research Center, MS 235A, Hampton, VA 23665-5225; (804) 865-2791.

NATO Advanced Study Institute on Testing and Diagnosis of VLSI and ULSI, June 22-July 4, Como, Italy. Contact F. Lombardi, University of Colorado at Boulder, Campus Box 425, Boulder, CO 80309; (303) 492-1437 or M. Sami, Dept. di Elettronica, Politecnico di Milano, Piazza Leonardo da Vinci 32, I-20133 Milan, Italy; phone 39 (02) 2399-3516.

Second International Conference on Computers and Applications, June 23-27, Beijing. Contact Tse-yun Feng, Penn State University, Electrical Engineering East Bldg., University Park, PA 16802; (814) 863-1469 or Oscar N. Garcia, Dept. of Electrical Engineering and Computer Science, Rm. T-637, George Washington University, 801 22nd St. NW, Washington, DC 20052; (202) 676-7175.

NECC-87, Eighth National Educational Computing Conference (ACM, SCS), June 24-26, Philadelphia. Contact Frank L. Friedman, Computer Activities Bldg., Box JA1, Dept. of Computer and Information Sciences, Temple University, Philadelphia, PA 19122; (215) 787-8450.

SIGPlan 87, Symposium on Interpreters and Interpretive Techniques (Computer Society, ACM), June 24-26, St. Paul, Minnesota. Contact Mark Scott Johnson, Sun Microsystems, 2550 Granada Ave., Mountain View, CA 94043; (415) 857-8719.

Workshop on Group Technology and Computer-Automated Process Planning (SME), June 25, Hartford, Connecticut. Contact Society of Manufacturing Engineers, Technical Activities Dept., 1 SME Dr., PO Box 930, Dearborn, MI 48121; (313) 271-1080, ext. 365.

DAC-87, 24th ACM/IEEE Design Automation Conference, June 28-July 1, Miami Beach, Florida. Contact Design Automation Conference, P.O. Pistilli, MP Associates, 7366 Old Mill Trail, Boulder, CO 80301; (303) 530-4333.

Compass 87, Conference on Computer Assurance (IEEE), June 29-July 3, Washington, DC. Contact Frank Houston, PO Box 5314, Rockville, MD 20851; (301) 443-5020.

First International Conference on Industrial and Applied Mathematics (GAMM, IMA, INRIA, SIAM, SMAI), June 29-July 3, Paris. Contact SIAM, 117 S. 17th St., Philadelphia, PA 19103; (215) 564-2929.

July 1987

CAR-87, Second International Conference on Computer-Assisted Radiology (Computer Society, ACM), July 1-4, Berlin. Contact Michael L. Rhodes, Multi-Planar Diagnostic Imaging, Inc., 2730 Pacific Coast Hwy., Torrance, CA 90505; (213) 539-5944.

EDIF (Electronic Design Interchange Format) Workshop (Computer Society, ACM), July 2, Miami, Florida. Contact John Andrews, Fairchild Semiconductor, MS 7-26, 333 Western Ave., South Portland, ME 04106; (207) 775-8368.

Workshop on Real-Time Operating Systems, July 2-3, Cambridge, Massachusetts. Contact Krithi Ramamritham, Dept. of Computer and Information Science, Graduate Research Center, University of Massachusetts, Amherst, MA 01003; (413) 535-0196 or Al Mok, Dept. of Computer Science, University of Texas, Austin, TX 78713; (512) 471-9542.

Foundations of Computer-Aided Process Operations (AIChE, Cache Corp.), July 5-10, Park City, Utah. Contact Cache Corp., PO Box 7939, Austin, TX 78713, (512) 471-4933.

FTCS-17, 17th International Symposium on Fault-Tolerant Computing, July 6-8, Pittsburgh. Contact John Shen, Carnegie Mellon University,

Conferences that the Computer Society participates in or sponsors are indicated by the Computer Society logo; additional conference sponsors are listed in parentheses. Other conferences of interest to our readers are also included.

For inclusion in Calendar or Call for Papers, submit information **six weeks before the month of publication** (e.g., for the **September 1987** issue, send information for receipt by **July 15, 1987**) to Calendar Editor, *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720.

Dept. of Electrical and Computer Engineering, Pittsburgh, PA 15213; (412) 268-3601.

25th Annual Meeting of the Association for Computational Linguistics, July 6-9, Stanford, California. Contact Don Walker, Bell Communications Research, 445 South St., MRE 2A379, Morristown, NJ 07960; (201) 829-4312.

Directions and Implications of Advanced Computing, July 12, Seattle. Contact Computer Professionals for Social Responsibility, Box 85481, Seattle, WA 98105 or Doug Schuler, 3015 NW 58th, Seattle, WA 98107; (206) 865-3228.

AAAI-87, Sixth National Conference on Artificial Intelligence, July 13-17, Seattle. Contact American Association for Artificial Intelligence, 445 Burgess Dr., Menlo Park, CA 94025-3496.

1987 Summer Computer Simulation Conference (SCS), July 27-30, Montreal, Quebec, Canada. Contact Society for Computer Simulation, PO Box 17900, San Diego, CA 92117; (619) 277-3888.

ACM SIGGraph 87, July 27-31, Anaheim, California. Contact SIGGraph 87 Conference Management, Smith Bucklin and Associates, Inc., 111 E. Wacker Dr., Suite 600, Chicago, IL 60601; (312) 644-6610.

14th Annual Conference on Computer Graphics and Interactive Techniques (ACM), July 27-31, Anaheim, California. Contact SIGGraph 87 Conference Management, Smith Bucklin and Associates, Inc., Suite 600, Chicago, IL 60601; (312) 644-6610.

10th IFAC World Congress, July 27-31, Munich, West Germany. Contact IFAC Secretariat, Schlossplatz 12, A-2361 Laxenburg, Austria.

ICGATA-87, Second International Conference on Genetic Algorithms and Their Applications (AAAI, US Navy Center for Applied Research in AI), July 28-31, Cambridge, Massachusetts. Contact John J. Grefenstette, Navy Center for Applied Research in AI, Code 5510, Naval Research Laboratory, Washington, DC 20375-5000.

August 1987

25th Annual Conference of the Urban and Regional Information Systems Association (URISA), August 2-6, Fort Lauderdale, Florida. Contact Thomas Palmerlee, URISA, 319 C St. SE, Washington, DC 20003; (202) 543-7141.

Seventh International Conference on Computer Science (IEEE), August 3-7, Santiago, Chile. Contact Hector Garcia-Molina, Dept. of Computer Science, Princeton University, Princeton, NJ 08544.

12th Structured Methods Conference, August 3-7, Grand Rapids, Michigan. Contact Derek Hatley, Lear Siegler, Inc., Instrument Division, 4141 Eastern Ave. SE, Grand Rapids, MI 49508.

1987 International Conference on Chinese Information Processing, August 4-6, Beijing. Contact Fan Bing Jung, China Computer Technical Service Corp., PO Box 4708, Beijing, China.

IEEE/ACM Symposium on the Simulation of Computer Networks (Computer Society, ACM, SCS), August 5-7, Colorado Springs, Colorado. Contact Mitchell Spiegel, GTE Systems, 1700 Research Blvd., Rockville, MD 20850; (301) 294-8400.

1987 ASME International Computers in Engineering Conference and Exhibition, August 9-13, New York. Contact T. James Cokonis, General Electric Co., Advanced Development and Information Systems, PO Box 8555, Bldg.7, Rm. 7244, Philadelphia, PA 19101; (215) 354-5112.

Sixth ACM SIGACT-SIGOps Symposium on Distributed Computing, August 10-12, Vancouver, British Columbia, Canada. Contact David Kirkpatrick, Dept. of Computing Science, University of British Columbia, Vancouver, British Columbia, Canada, or Fred B. Schneider, Dept. of Computer Science, Upson Hall, Cornell University, Ithaca, NY 14853.

Sixth International Conference on Mathematical Modeling: An Interdisciplinary Integrative Forum on Mathematical Modeling in Engineering, Economics, Biological, Medical, Environmental, Social, and Other Sciences, August 10-13, St. Louis. Contact Ervin Y. Rodin, Dept. of Systems Science and Mathematics, Washington University, Box 1040, St. Louis, MO 63130; (314) 889-5806.

International Conference on Human-Computer Interaction, August 10-15, Honolulu. Contact Gavriel Salvendy, School of Industrial Engineering, Purdue University, West Lafayette, IN 47907; (317) 494-5426.

Crypto 87, August 16-20, Santa Barbara, California. Contact Gordon B. Agnew, Dept. of Electrical Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada; phone (519) 885-1211, ext. 3041.

Engineering and Manufacturing 87 (NCGA), August 17-20, Boston. Contact Christine Radiske or Marjorie Foos, National Computer Graphics Association, 2722 Merrilee Dr., Suite 200, Fairfax, VA 22031; (703) 698-9600.

Ninth International Conference on Production Research (Computer Society, IFPR), August 17-20, Cincinnati, Ohio. Contact Ernest L. Hall, Center for Robotics Research, University of Cincinnati, ML 72, Cincinnati, OH 45221.

1987 International Conference on Parallel Processing, August 17-21, St. Charles, Illinois. Contact T. Feng, Pennsylvania State University, Dept. of Electrical Engineering, University Park, PA 16802; (814) 863-1469.

IJCAI-87, 10th International Joint Conference on Artificial Intelligence, August 23-28, Milan, Italy. Contact Marco Somalvico, Dipartimento di Elettronica, Politecnico di Milano, Piazza Leonardo Da Vinci N.32, I-20133 Milan, Italy, or John McDermott, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213; (412) 268-2599.

International Workshop on Petri Nets and Performance Models (Computer Society, ACM), August 24-26, Madison, Wisconsin. Contact Tadao Murata, University of Illinois at Chicago, Dept. of Electrical Engineering and Computer Science, MC 154, Box 4348, Chicago, IL 60680; (312) 996-2307 or (312) 996-3422.

1987 IEEE Workshop on Languages for Automation, August 24-27, Vienna. Contact Shi-Kuo Chang, Dept. of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260; (412) 624-8441.

Eurographics 87 (ACM, IFIP), August 24-28, Amsterdam. Contact Secretariat Eurographics 87, c/o Organisatie Bureau Amsterdam, Europaplein 12, 1078 GZ Amsterdam, The Netherlands; phone 31 (20) 44-08-07.

Tencon, August 26-28, Seoul, South Korea. Contact Jung Uck Seo, Korea Telecom Authority, 100 Sejong-Re, Chongro-ku, Seoul 110, South Korea.

1987 Annual International Test Conference, August 30-September 4, Washington, DC. Contact Doris Thomas, PO Box 264, Mount Freedom, NJ 07970; (201) 895-5260.

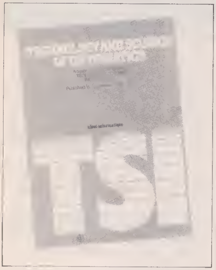
Topical Meeting on Artificial Intelligence and Other Innovative Computer Applications in the Nuclear Industry, August 31-September 2, Snowbird, Utah. Contact John I. Sackett, Argonne National Laboratory, PO Box 2528, Idaho Falls, ID 83403; (208) 526-7214.

Symposium on Logic Programming, August 31-September 4, San Francisco. Contact David S. Warren, Quintus Computer Systems, Inc., 1310 Villa St., Mountain View, CA 94041.

September 1987

Second IFIP Conference on Human Interaction, September 1-4, Stuttgart, West Germany. Contact H.J. Bullinger, FhG IAO, Holzgarteustr. 17, 7000 Stuttgart 1, West Germany.

Now published by John Wiley & Sons



The English edition of the leading French computing journal ***Technique et Science Informatiques (TSI)*** *is now published by Wiley.*

Covering computer science at the very highest level ***TSI*** *brings you reports on the latest developments across a broad spectrum of research.*

Subjects regularly featured include Artificial Intelligence, Computer-aided Design and Manufacturing, Information Systems, Modelling and Simulation, Office Automation and Software Engineering.

For ease of reference each issue is presented in three distinct sections —
SURVEY contains reviews that will keep the non-specialist fully informed of progress in all areas
APPLICATIONS describes results of practical experiments using known concepts
RESEARCH presents original articles concerned with the latest advances in information science.

Send now for your complimentary copy.

Volume 6 (1987) 9 issues
$225.00 (UK £120.00)
Individual subscribers can save 25% with a personal subscription.

Please send me a complimentary copy of TSI

Name ....

Organization ....

Address ....

....

....

....

Return this coupon to either of these addresses —

Dept AC, John Wiley & Sons Ltd. Baffins Lane, Chichester, Sussex PO19 1UD, England

Subscription Dept C, John Wiley & Sons Inc, 605 Third Avenue, New York, NY 10158, USA

Reader Service Number 6

13th International Conference on Very Large Databases (Computer Society, ACM, BCS), September 1-4, Brighton, England. Contact Stuart E. Madnick, Massachusetts Institute of Technology, Sloan School of Management, 50 Memorial Dr., E53-317, Cambridge, MA 02139; (617) 253-6671.

Summer Conference on Category Theory and Computer Science, September 7-9, Edinburgh, Scotland. Contact George Cleland, Laboratory for Foundations of Computer Science, Dept. of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, Scotland, UK.

European Software Engineering Conference (BCS), September 7-10, Strasbourg, France. Contact John Buxton, Dept. of Computing, King's College, University of London, Strand, London WC2R 2LS, England, UK.

HCI-87, Conference on People and Computers (BCS), September 7-11, Exeter, England. Contact HCI-87 Conference, B.I.S.L., 13 Mansfield St., London W1M 0BP, England, UK.

Third International Conference on Functional Programming Languages and Computer Architecture (ACM, IFIP, INRIA), September 14-16, Portland, Oregon. Contact John H. Williams, IBM Almaden Research Center, K53/803, 650 Harry Rd., San Jose, CA 95120-6099.

Euromicro 87, 13th Symposium on Microprocessing and Microprogramming: Microcomputers—Usage, Methods, and Structures, September 14-17, Portsmouth, England. Contact Chiquita Snippe-Marlisa, p/a TH Twente, gebouw TW/RC, Rm. A227, PO Box 217, 7500 AE Enschede, The Netherlands; phone 31 (53) 33-87-99.

ICDCS-7, Seventh International Conference on Distributed Computing Systems, September 14-18, Berlin. Contact R. Popescu-Zeletin, Hahn-Meitner-Institut Berlin, Glienicker Strasse 100, D-1000, Berlin 39, West Germany; phone 49 (30) 8009-2594 or 49 (30) 8009-2541.

ICCC-ISDN 87, Evolving to Integrated Services Digital Networks in North America, September 15-17, Dallas. Contact Caroline Stites, Bell Atlantic, 1310 N. Court House Rd., tenth floor, Arlington, VA 22201; (703) 974-5453.

Midcon 87 (IEEE), September 15-17, Rosemont, Illinois. Contact Alexes Razevich, Electronic Conventions Management, 8110 Airport Blvd., Los Angeles, CA 90045; (213) 772-2965 or (800) 421-6816.

Second Workshop on Artificial Intelligence Applications in Environmental Sciences (NOAA/ERL), September 15-17, Boulder, Colorado. Contact William Moninger, NOAA/ERL, R/E2, 325 Broadway, Boulder, CO 80303.

26th Lake Arrowhead Workshop: Specifying Concurrent Systems in the Year 2000, September 16-18, Lake Arrowhead, California. Contact Brent Hailpern, IBM T.J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598; (914) 789-7797; CSnet bth@ibm.com.

CSM-87, Conference on Software Maintenance (Computer Society, AWC, DPMA, NBS, SMA), September 21-24, Austin, Texas. Contact Roger Martin, National Bureau of Standards, Bldg. 225, Rm. B266, Gaithersburg, MD 20899; (301) 921-3545 or Computer Society of the IEEE, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903; (202) 371-0101.

1987 Design Automation Conference (ASME), September 27-30, Boston. Contact S.S. Rao, School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907; (317) 494-5699.

International Conference on Software Engineering for Real-Time Systems, September 28-30, Cirencester, England. Contact R. Larry, Institute of Electronic and Radio Engineers, 99 Gower St., London WC1E 6AZ, England UK.

Oceans 87, September 28-October 1, Halifax, Nova Scotia, Canada. Contact Alan R. Longhurst, Bedford Institute of Oceanography, PO Box 1006, Dartmouth, Nova Scotia B2Y 4A2, Canada.

Fifth International Symposium on Data Analysis and Informatics (INRIA), September 29-October 2, Versailles, France. Contact INRIA, Service des Relations Exterieures, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France.

25th Annual Allerton Conference on Communication, Control, and Computing, September 30-October 2, Monticello, Illinois. Contact P.R. Kumar or Michael C. Loui, University of Illinois at Urbana-Champaign, Coordinated Science Laboratory, 1101 W. Springfield Ave., Urbana, IL 61801.

October 1987

AAAI Workshop on Spatial Reasoning and Multisensor Fusion, October 5-7, St. Charles, Illinois. Contact Su-shing Chen, Dept. of Computer Science, University of North Carolina, Charlotte, NC 28223; (704) 547-4880.

IFIP Conference on Distributed Processing, October 5-7, Amsterdam. Contact M.H. Barton, Electrical Engineering, University of Bristol, BS8 1TR, England, UK.

1987 Workshop on Computer Architecture for Pattern Analysis and Machine Intelligence, October 5-7, Seattle.

Contact Steve Tanimoto, Dept. of Computer Science, FR-35, University of Washington, Seattle, WA 98195; (206) 543-1695.

10th Data Communications Symposium, October 5-7, Napa, California. Contact Bart Stuck, Probe Research, PO Box 590, Morristown, NJ 07960; (201) 285-1500.

12th Conference on Local Computer Networks, October 5-7, Minneapolis, Minnesota. Contact Stephane Johnson, Start, Inc., 10301 Toledo Ave. South, Bloomington, MN 55437; (612) 831-2122.

ASPLOS-II, Second International Conference on Architectural Support for Programming Languages and Operating Systems (Computer Society, ACM), October 5-8, Palo Alto, California. Contact Randy H. Katz, Computer Sciences Division, UC Berkeley, Evans Hall, Berkeley, CA 94720 or Martin Freeman, Signetics Corp., 811 E. Arques Ave., Sunnyvale, CA 94086; (408) 991-3591.

ICCD-87, IEEE International Conference on Computer Design: VLSI in Computers and Processors, October 5-8, Rye Brook, New York. Contact Prathima Agrawal, AT&T Bell Laboratories, 600 Mountain Ave., Rm. 3D-480, Murray Hill, NJ 07974; (201) 582-6943.

IWDM-87, Fifth International Workshop on Database Machines (ACM), October 5-8, Karuizawa, Nagano Prefecture, Japan. Contact M. Kitsuregawa, Institute of Industrial Science, University of Tokyo, 7-22-1, Roppongi, Minato-ku, Tokyo 106, Japan; phone 81 (03) 402-6231.

OOPSLA-87, Object-Oriented Programming Systems, Languages, and Applications (ACM), October 5-8, Kissimmee, Florida. Contact Chet Winiski, Productivity Products International, 27 Glen Rd., Sandy Hook, CT 06482.

AAAIC-87, Aerospace Applications of Artificial Intelligence Conference, October 5-9, Dayton, Ohio. Contact Michael Johnston, BDM Corp., 1900 Founders Dr., Kettering, OH 45420; (513) 259-4434.

Compsac 87 (Computer Society, IPSJ), October 5-9, Tokyo. Contact Tosiyasu L. Kunii, c/o Business Center for Academic Societies Japan, Yamazaki Bldg. 4F, 2-40-14, Hongo, Bunkyo-ku, Tokyo 113, Japan; phone 81 (3) 817-5831, or Albert K. Hawkes, Sargent & Lundy, Engineering Consultants, 55 E. Monroe, Chicago, IL 60603; (312) 269-3640, or Stephen S. Yau, Northwestern University, Dept. of Electrical Engineering and Computer Science, Evanston, IL 60201; (312) 491-3641.

Computers and Aerospace VI Conference (ACM), October 7-9, Boston. Contact Malcolm Steifel, MS A-160, Mitre Corp., Bedford, MA 01730.

International Workshop on AI Applications to CAD Systems for Electronics (IEEE), October 8-10, Munich, West Germany. Contact Werner Sammer, Siemens AG, Corporate Research and Technology, Otto-Hahn-Ring 6, 8000 Munich 83, West Germany; phone 49 (89) 636-3348 or Alfred C. Weaver, Lockheed EMSCO, MS B-08, 2400 NASA Rd. One, Houston, TX 77058; (713) 333-6792.

Astronomy from Large Databases: Scientific Objectives and Methodological Approaches, October 12-14, Garching-bei-Munchen, West Germany. Contact F. Murtagh, ST-ECF/ESO, Karl Schwarzschild-Str. 2, D-8046 Garching-bei-Munchen, West Germany.

FOCS-87, October 12-16, Los Angeles. Contact Ashok Chandra, IBM T.J. Watson Research Center, PO Box 218, Yorktown Heights, NY 10598; (914) 945-1752.

International Symposium on Methodologies for Intelligent Systems, October 14-17, Charlotte, North Carolina. Contact Zbigniew W. Ras, Dept. of Computer Science, University of North Carolina, Charlotte, NC 28223; (704) 547-4567.

Third Annual Expert Systems in Government Conference (Computer Society, AIAA), October 19-23, Washington, DC. Contact Peter Bonasso, Mitre Washington AI Center, 7725 Colshire Blvd., MS W952, McLean, VA 22102; (703) 883-6908.

Second International Conference on Data and Knowledge for Manufacturing and Knowledge (ACM), October 19-24, Hartford, Connecticut. Contact Fred Maryanski, University of Connecticut, Computer Science and Engineering Dept., U-155, Storrs, CT 06268.

FJCC-87, Fall Joint Computer Conference (Computer Society, ACM), October 25-29, Dallas. Contact Debra Anthony, Texas Instruments, 6500 Chase Oaks Blvd., PO Box 86905, MS 8419, Plano, TX 75086; (214) 575-2151.

Government Microcircuits Applications Conference, October 27-29, Orlando, Florida. Contact Frank J. Rehm, RADX Griffiss Air Force Base; (315) 330-7781.

CCDC-87, Computer Communications for Developing Countries, October 27-30, New Delhi, India. Contact S. Ramani, National Centre for Software Technology, Gulmohar Cross Road No. 9, Juhu, Bombay 400049, India; phone 629574 or 629606.

November 1987

11th Symposium on Computer Applications in Medical Care, November 1-4, Washington, DC. Contact SCAMC Secretariat, George Washington University Medical Center, Office of Continuing Medical Education, 2300 K St. NW, Washington, DC 20037.

Third Annual Conference on Artificial Intelligence for Space Applications (NASA), November 2-3, Huntsville, Alabama. Contact Thomas S. Dollman, NASA/EB44, Marshall Space Flight Center, AL 35812; (205) 544-3823.

21st Annual Asilomar Conference on Signals, Systems, and Computers (IEEE, Naval Postgraduate School), November 2-4, Pacific Grove, California. Contact Douglas F. Elliott, Rockwell International Corp., 3370 Miraloma Ave., MS BD07, Anaheim, CA 92803-3170.

Workshop on Workstation Operating Systems, November 5-6, Cambridge, Massachusetts. Contact Luis-Felipe Cabrera, 6572 Northridge Dr., San Jose, CA 95120; (408) 927-1838.

11th ACM Symposium on Operating Systems Principles, November 9-11, Austin, Texas. Contact Alfred Spector, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 or Les Belady, MCC, 9430 Research Blvd., Echelon Bldg. No. 1, Suite 200, Austin, TX 78759; (512) 834-3330.

ICCAD-87, IEEE International Conference on Computer-Aided Design, November 9-12, Santa Clara, California. Contact Basant Chawla, AT&T Bell Laboratories, 1247 S. Cedar Crest Blvd., Allentown, PA 18103; (215) 770-3484.

International Conference on Information Science and Engineering (Computer Society, IERE), November 25-27, York, England. Contact R. Larry, Institute of Electronic and Radio Engineers, 99 Gower St., London WC1E 6AZ, England, UK; 44 (01) 388-3071.

Workshop on Computer Vision, November 30-December 2, Miami Beach, Florida. Contact Harry Hayman, 738 Whittaker Terrace, Silver Spring, MD 20901; (301) 434-1990.

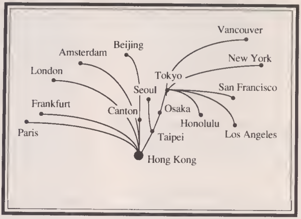
December 1987

Eighth Real-Time Systems Symposium, December 1-3, San Francisco. Contact Kang G. Shin, Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-1109; (313) 763-0391.

Performance 87 (ACM, AFCET, IFIP), December 7-9, Brussels. Contact G. Latouche, Universite Libre de Bruxelles, Campus Plain CP212-bd du Triomphe, 1050 Brussels, Belgium.

1987 FAR EAST COMPUTER AND SOFTWARE TECHNOLOGY TOUR

DEVELOP NEW PRODUCTS AND MARKETS IN THE FAR EAST



Hundreds in the electronics, computer, and software industry have taken advantage of Commerce Tours International's tours to the Orient. This year—our 11th year in the business—you, too, can develop new products, markets, and sources in the Far East. Develop contacts with future business partners in the Orient. Meet your foreign competitors in the Pacific Rim. Gather ideas. Exchange observations. It's truly a business opportunity you won't want to miss!

DATE: October 3-16, 1987

VISIT: Japan, Taiwan, Korea, and *optional extensions:*

A: Hong Kong
B: Canton and Beijing, China

PROGRAM HIGHLIGHTS:

VISIT ELECTRONICS SHOWS

- Japan Software Show
- Japan Electronics Show
- Korea Electronics Show
- Taiwan Electronics Show
- Canton Trade Fair

ATTEND BUSINESS SEMINARS

- Japan Electronics Technology Update
- Asian Business Convention (Osaka, Japan)
- Korea Business Opportunities Seminar
- Taiwan Business Opportunities Seminar
- China Business Opportunities Seminar

ATTEND COMPSAC 87 TOKYO

The Computer Society of the IEEE's 11th Annual Computer Software and Applications Conference

DEPARTURE CITIES:

Departures will be available from major U.S., Canadian, and European cities via regular flights.

PRICE:

$3,130 from the West Coast, including airfare, hotel accommodations, daily American breakfast, five dinners, transportation to/from show grounds, airport/hotel transfers, and all business programs as outlined above. Japan update seminar, Compsac conference, and Asian business convention registration fees are additional.

ORGANIZED BY:

Commerce Tours International, Inc.

FOR MORE INFORMATION, PLEASE CONTACT ONE OF THE FOLLOWING OR RETURN THIS COUPON

NAME ______________________
COMPANY ______________________
DEPARTMENT/TITLE ______________________
ADDRESS ______________________
CITY ____________ STATE ______ ZIP ________
TELEPHONE ______________________
(Computer/Software Tour)

IN U.S.A

Commerce Tours International
870 Market Street
Suite 920
San Francisco, CA 94102
Tel: (415) 433-3072
Attn.: Bob Chang

IN CANADA

Plesman Publications
2 Lansing Square, #703
Willowdale, ONT M2J 4P8
Tel: (416) 497-9562
Attn.: Sharon Raspin

IN U.K.

JCT International Ltd.
Hanover Court
5 Hanover Square
London WIR 9 HE
Tel: 01-499-6827
Attn.: Gordon MacFarland

IN NETHERLANDS

Nippon Express (Nederland) B.V.
Postbox 7175
1007 JD Amsterdam
Tel: 020-621387
Attn.: Errol Lafleur

Reader Service Number 7

CALL FOR PAPERS

Call for papers for Computer

Computer magazine seeks articles that cover the state of the art and important new developments in computer science, technology, and applications. Aimed at a broad audience with diverse interests and experience, *Computer* usually publishes surveys or tutorials that facilitate the transfer of technology from university to industry, from research to applications, and across specialized fields. Submit six copies of the manuscript, including illustrations, references, and authors' biographies, to

- Bruce Shriver
  IBM T.J. Watson Research Center
  Route 134
  PO Box 704, H0-B04A
  Yorktown Heights, NY 10598
  (914) 789-7626
  Compmail+: b.shriver
  CSnet: shriver@ibm.com
  Vnet: shriver at yktvmh

Computer also seeks contributions for a special issue on integrated optics for computing and signal processing. The focus of this special issue will be integrated optical devices and architectures for optical processing, as opposed to two-dimensional bulk optical technologies. The one-dimensional integrated optical technology makes use of devices fabricated with conventional integrated electronic techniques and with electronic materials, such as GaAs. Papers are solicited that discuss the use of integrated optical devices, circuits, and architectures for signal processing and optical computing.

Six copies of the manuscript should be submitted by **June 30, 1987,** to T.E. Batchman, who is serving as co-guest editor with E.A. Parrish, Jr. He can be reached at the Dept. of Electrical Engineering, Vanderbilt University, Charlottesville, VA 22901; (804) 924-3960.

Review will be conducted promptly and authors will be notified of the results by mid-August. Publication of the special issue is scheduled for December 1987.

Computer Society of the IEEE Technical Committee on Computer Education: Contributions up to five typewritten pages are welcomed for the TCCE newsletter, a forum for the exchange of ideas among persons interested in computer education or computers in education. Direct news items, short articles, and any correspondence to Helen Hays, Dept. of Computer Science, Southeast Missouri State University, Cape Girardeau, MO 63701; (314) 651-2244.

International Journal for Artificial Intelligence in Engineering: Papers are sought, particularly those with emphasis on research and development leading to problem-solving strategies. For information and submission requirements, contact D. Sriram, Dept. of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 or K.J. MacCallum, Dept. of Ship and Marine Technology, Marine Technology Center, 100 Montrose St., Glasgow, Scotland, UK.

International Journal of Pattern Recognition and Artificial Intelligence: Submit four copies of manuscripts to H. Bunke, Universitat Bern, Institut fur Informatik und Angewandte Mathematik, Langgasstrasse 51, CH-3012 Bern, Switzerland, or Patrick Wang, College of Computer Science, Northeastern University, 360 Huntington Ave., Boston, MA 02115; (617) 437-3711.

1988 ACM SIGMetrics Conference on Measurement and Modeling of Computer Systems: May 24-27, 1988, Santa Fe, New Mexico. Abstracts of papers are due in **July 1987,** and six copies of individual papers will be due by **September 30, 1987.** Extended abstracts for poster sessions are due **October 30, 1987.** Proposals for tutorial sessions are also sought. Submit materials to Kenneth C. Sevcik, Computer Science Dept., University of Toronto, Toronto M5S 1A4, Ontario, Canada; phone (416) 978-6219.

Distributed Simulation Conference (part of the 1988 SCS Multiconference): February 3-5, 1988, San Diego, California. Submit extended abstracts (at least 2000 words) by **July 1, 1987,** to Brian W. Unger, SCS, PO Box 17900, San Diego, CA 92117; (619) 277-3888.

Seventh Annual IEEE Phoenix Conference on Computers and Communications: March 16-18, 1988, Scottsdale, Arizona. Submit five copies of a complete paper (5000 words maximum) that includes an abstract of 200 to 300 words by **July 3, 1987,** to Alex C. Brown, Jr., Goodyear Aerospace Corp., MS 1212, PO Box 85, Litchfield Park, AZ 85340; (602) 925-7509. Persons interested in submitting proposals for panel sessions should contact Alex Brown. Persons interested in offering tutorials should contact Forouzan Golshani, Dept. of Computer Science, Arizona State University, Tempe, AZ 85287; (602) 965-2855.

Second Workshop on Artificial Intelligence Applications in Environmental Sciences (NOAA/ERL): September 15-17, 1987, Boulder, Colorado. Submit an abstract (300 words maximum) by **July 15, 1987,** to William Moninger, NOAA/ERL, R/E2, 325 Broadway, Boulder, CO 80303.

Ninth International Wroclaw Symposium on Electromagnetic Compatibility: June 28-30, 1988, Wroclaw, Poland. Submit an abstract (50 to 75 words) and a summary (500 to 750 words) by **July 15, 1987,** to F.L. Stumpers, Elzentlaan 11, Eindhoven 5611 L.G., The Netherlands. Submit four copies of the abstract and four copies of the summary by the same date to EMC Symposium, Box 2141, 51-645 Wroclaw 12, Poland.

25th Annual Allerton Conference on Communication, Control, and Computing: September 30-October 2, 1987, Monticello, Illinois. Submit the title and a 1000-word summary of a paper suitable for a 20-minute presentation, or the title and a 500-word summary of a paper suitable for a 10-minute presentation by **July 27, 1987,** to P.R. Kumar, University of Illinois at Urbana-Champaign, Coordinated Science Laboratory, 1101 W. Springfield Ave., Urbana, IL 61801.

Conferences that the Computer Society participates in or sponsors are indicated by the Computer Society logo; additional conference sponsors are listed in parentheses. Other conferences of interest to our readers are also included.

For inclusion in Call for Papers or Calendar, submit information **six weeks before the month of publication** (e.g., for the **September 1987** issue, send information for receipt by **July 15, 1987**) to Calendar Editor, *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720.

Workshop on Workstation Operating Systems: November 5-6, 1987, Cambridge, Massachusetts. Potential participants should submit 10 copies of a position statement describing their experiences, interests, and future directions that are related to the workshop topic by **July 27, 1987,** to Luis Felipe Cabrera, IBM Almaden Research Center, 650 Harry Rd., San Jose, CA 95120-6099; (408) 927-1838.

Second IEEE Conference on Computer Workstations: March 7-10, 1988, Santa Clara, California. Submit six copies of the paper (5000 words maximum) by **July 29, 1987,** to Lawrence Stewart, DEC Systems Research Center, 130 Lytton Ave., Palo Alto, CA 94301.

Third Annual Conference on Artificial Intelligence for Space Applications (NASA): November 2-3, 1987, Huntsville, Alabama. Submit unclassified abstracts not exceeding 400 words by **July 31, 1987,** to Thomas S. Dollman, NASA/EB44, Marshall Space Flight Center, AL 35812; (205) 544-3823.

IEEE Software: Materials on fourth-generation language development, software legal aspects, and software modeling are sought for the March 1988 issue. Contact Ted Lewis, editor-in-chief, *IEEE Software,* c/o Computer Science Dept., Oregon State University, Corvallis, OR 97331; (503) 754-3273; CSnet, lewis@oregon-state; Compmail+, t.lewis. Materials are due by **August 1, 1987.**

Information Sciences—An International Journal: Papers are sought for a special issue to be published in early 1988 on database systems. Submit four copies of papers by **August 1, 1987,** to Ahmed Elmagarmid, Computer Engineering Program, 121 Electrical Engineering East Bldg., Pennsylvania State University, University Park, PA 16802; (814) 863-1047.

IEEE Infocom 88: Networks—Evolution or Revolution?: March 28-31, 1988, New Orleans. Submit four copies of the complete paper by **August 3, 1987,** to Al Leon-Garcia, Dept. of Electrical Engineering, University of Toronto, Toronto, Ontario M5S 1A4, Canada; phone (416) 978-5037.

CompEuro 88: System Design—Concepts, Methods, and Tools: April 11-14, 1988, Brussels. Papers on theoretical and practical aspects of system design are sought. Submit five copies of the complete paper (20 double-spaced pages maximum) by **August 15, 1987,** to Pierre Wodon, Philips Research Laboratory Brussels, 2 Ave. Van Becelaere, bte 8, B-1170 Brussels, Belgium; phone 32 (02) 673-41-90.

Control 88 (IEE): April 13-15, 1988, Oxford, England. Submit an extended synopsis (1500 words maximum) by **August 28, 1987,** to Conference Services, IEE, Savoy Pl., London WC2R 0BL, England, UK; phone 44 (01) 240-1871.

International Workshop on Robot Control: Theory and Applications (IEE): April 11-12, 1988, Oxford, England. Submit a synopsis (1500 words maximum) by **August 28, 1987,** to the Computing and Control Division, IEE, Savoy Pl., London WC2R 0BL; phone 44 (01) 240-1871, ext. 330.

1988 Eastern Simulation Conferences: Simulators V, Tools for the Simulationist, the Simulation Profession, Credibility Assessment, and Simulation Languages (SCS): April 18-21, 1988, Orlando, Florida. Send one-page abstracts and session proposals by **August 31, 1987,** to Eastern Simulation Conferences, SCS, PO Box 17900, San Diego, CA 92117. Proposals for professional development seminars are also sought; these should be sent to SCS.

International Symposium and Exposition on Robots: November 6-10, 1988, Sydney, Australia. Submit a title and a 500-word abstract by **September 1, 1987,** to R.A. Jarvis, International Symposium and Exposition on Robots, GPO Box 1527, Sydney, NSW 2001, Australia.

Second Conference on Applied Natural Language Processing (Association for Computational Linguistics): February 9-12, 1988, Austin, Texas. Submit 10 copies of a summary that is six to eight pages long to Bruce Ballard, 3C-440A, AT&T Bell Laboratories, Murray Hill, NJ 07974; (201) 582-5440. Materials must be received by **September 1, 1987.**

12th IMACS World Congress on Scientific Computation: July 18-22, 1988, Paris. This conference will devote a one-day session to second-generation expert systems. Papers are invited that describe expert systems that reason with deep knowledge. Emphasis will be put on work describing cooperation between heuristic and deep reasoning. Submit three copies of a 1000-word abstract by **September 1, 1987,** to Jean-Marc David, Laboratoires de Marcoussis, Computer Science Division, Route de Nozay, 91460, Marcoussis, France.

Computer Networking Symposium: April 11-13, 1988, Arlington, Virginia. Submit four copies of the complete paper or of a 1000-word extended abstract by **September 15, 1987,** to Jeffrey Jaffe, IBM Research Center, PO Box 704, Yorktown Heights, NY 10598.

IEEE International Conference on Robotics and Automation: April 25-29, 1988, Philadelphia. Papers 15 to 20 pages long are sought, as are papers five to seven pages long. Submit papers in either category by **September 15, 1987,** to Robert B. Kelley, ECSE Dept., Rensselaer Polytechnic Institute, Troy, NY 12180-3590.

COIS-88, Conference on Office Information Systems: March 23-25, 1988, Palo Alto, California. Send five copies of papers by **September 21, 1987,** to Robert B. Allen, 2A-367, Bell Communications Research, Morristown, NJ 07960; (201) 829-4315.

Fourth IEEE Conference on Artificial Intelligence Applications: March 14-18, 1988, San Diego, California. Full-length papers (5000 words maximum) and poster-session papers (1000 words maximum) are sought. Submit four copies of the full-length paper (include a 100-word abstract) by **September 25, 1987,** to Elaine Kant or Dennis O'Neill, Schlumberger-Doll Research, Old Quarry Rd., Ridgefield, CT 06877-4108. Submit four copies of poster-session papers to Elaine Kant or Dennis O'Neill by **December 21, 1987.**

Fourth International Conference on Pattern Recognition (BPRA, IAPR): March 28-30, 1988, Cambridge, England. Submit three copies of the complete paper (4000 words maximum) by **September 30, 1987,** to J. Kittler, Dept. of Electronic and Electrical Engineering, University of Surrey, Guildford GU2 5XH, England, UK.

14th International Conference on Electric Contacts (IEEE): June 20-24, 1988, Paris. Submit an abstract (200 words maximum) by **October 1, 1987,** to S.E.E., 48 Rue de la Procession, 75724 Paris Cedex 15, France.

1988 International Computers in Engineering Conference: Real-World Applications of Expert Systems and Artificial Intelligence (ASME): August 7-11, 1988, San Francisco. Submit abstracts by **October 15, 1987,** to Edward M. Patton, US Army Ballistic Research Lab, Aberdeen Proving Grounds, MD 21005.

IEEE Transactions on Computers: Papers are sought for a special issue on architectural support for programming languages and operating systems. Guidelines for submitting manuscripts appear in every issue of *IEEE Transactions on Computers.* Send seven copies of the manuscript by **November 1, 1987,** to Randy Katz, Dept. of Electrical Engineering and Computer Science, Computer Science Division, Evans Hall, University of California, Berkeley, CA 94720; (415) 642-8778.

The Role of Artificial Intelligence in Databases and Information Systems (IFIP): July 4-8, 1988, Canton, China. Submit five copies of the paper (5000 words maximum) by **November 2, 1987,** to Robert Meersman, Infolab, K.U. Brabant, Postbus 90153, NL-5000 Le Tilburg, The Netherlands; phone 31 (13) 66-24-30.

Coling 88, 12th International Conference on Computational Linguistics: August 22-27, 1988, Budapest, Hungary. Submit four copies of an extended abstract (seven pages maximum) and of a title page that includes a five-line summary and a specification of the topic area by **December 10, 1987,** to Eva Hajicova, Charles University, Faculty of Mathematics/Linguistics, Malostranske n. 25, 118 00 Prague 1, Czechoslovakia.

Call for Participants

10th Data Communications Symposium Building the Global Network

Napa Valley, California 5-7 October 1987

Sponsored by:

Association for computing machinery special interest group on Data Communications (SIGCOMM)

IEEE Computer Society Technical Committee on Computer Communications

IEEE Communications Society Technical Committees on Data Communication Systems, Computer Communications

Symposium Overview

The 10th Data Communications Symposium will focus on how to extend the current generation of data networks into a network of global proportion. There are many fundamental problems in naming, routing, protection, and other areas that need new insight to make a global network feasible. The symposium will be held as a workshop with the goal of bringing together a small group of experts in data communications to discuss these and other issues related to building the global network:

***Global Network Architecture:** What is needed in a global network that we do not have today? Will the global network be a single monolithic network like today's phone system or will it be a collection of loosely coupled autonomous networks? Will long-haul facilities remain costly and have lower bandwidth when compared to local networks? Where are administrative boundaries necessary and what form should they take?*

***Integrated End-User Services:** What services will users actually want from next generation networks? Who will be the consumers of these services: scientific supercomputing, information publishers, financial services? Is the integration of voice, data, and video a pipe dream? How will ISDN and FDDI interact?*

***Naming and Directories:** What is the global network's "phone book" going to look like? How will a directory service that effectively scales this problem to size be designed? Does the whole name structure (from top to bottom) have to be designed before work on a directory is begun?*

***Routing:** Are there stable and robust routing algorithms that can deal with global networks? In fact, what is the role of addressing in routing? How dynamic should routing be across separately managed regions of the network? Will the global network be connectionless or connection-oriented?*

***Protection:** Should the network or its endpoints provide authentication and access control? How will the network deal with various security models? Who will mediate among the multiple authentication domains and security perimeters?*

***Past Experiences:** What has been learned from past attempts to build community and enterprise-oriented networks? How do today's problems encountered by networks such as NSFnet and Physicsnet presage the problems of tomorrow? How will researchers and the standards communities cooperate to build the global network? Can (and must) everything be standardized?*

The workshop will be organized as a series of informal presentations and moderated panel discussions. To permit effective interaction, the workshop will be limited to approximately fifty people.

Program Committee

David Clark, Co-Chair, Massachusetts Institute of Technology

David Oran, Co-Chair, Digital Equipment Corporation

Vinton Cerf, Corporation for National Research Initiatives

David Mills, University of Delaware

Roger Needham, Cambridge University

Stephen Langdon, Amdahl Corporation

Participant Instructions

To attend, submit a 500 word (two page) written summary of your relevant work in this area, identifying the contribution that you are prepared to make at the workshop.

Send three copies of the proposal, a completed registration form, and a (refundable) $250 registration fee by June 15, 1987.

Attendees will be selected by the Program Committee on the basis of their written proposal. Notification of acceptance and rejection will be mailed by August 1, 1987. Accepted particiants will receive further information on the workshop and their participation. All other applicants will have their registration fee promptly returned.

10th Data Communications Symposium Registration Form

Registration Fee:
Make $250 check payable to 10th Data Communications Symposium

Mail three copies of proposal, registration form, and registration fee by June 15, 1987.

Send Registration Material To:
10th Data Communications Symposium
c/o David Clark
MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139
(617) 253-6003

Name ____________________

Address ____________________

Affiliation ____________________ *Telephone* ____________________

There will be no on-site or late registration for this workshop.

NEW PRODUCTS

Computer supports four operating systems

CIE Systems, a subsidiary of C. Itoh Electronics, offers a multiuser IBM PC-AT-compatible system that supports four operating systems: Pick, Unix, RM/COS, and MS-DOS.

The CIES/286 is media compatible with CIE Systems' line of 68020-based business systems. It supports two- to nine-user configurations.

The CIES/286 is based on Intel's 80286 processor running at 6, 8, or 10 MHz with memory expansion to 12M bytes. Storage components include 40 or 70M-byte internal hard-disk storage, 60M-byte 1/4-inch cartridge tape backup, and a 1.2M-byte AT-compatible floppy disk drive.

For graphics applications, the system can be fitted with an enhanced graphics adapter supporting IBM, CGA, MDA, and Hercules graphics and text modes.

Standard RS-232 cables and four-port communications controllers provide connection to ASCII terminals.

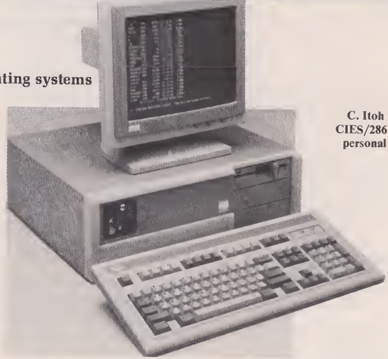
The CIES/286 comes in three standard models.

The CIES/2286 is a two-user system with a 40M-byte hard disk drive, floppy drive, 512K bytes of memory, and system console for $5960.

The CIES/6286 is a six-user system with a 70M-byte hard disk drive, 60M-byte streaming tape drive, floppy drive, 512K bytes of memory, and system console for $8085.

The CIES/9286 is a nine-user system with a 70M-byte hard disk drive, 60M-byte tape, floppy drive, 512K bytes of memory, and system console for $9800.

System prices vary according to the operating system used.



C. Itoh makes the CIES/286 multiuser personal computer system.

CIES/2286: **Reader Service 30**
CIES/6286: **Reader Service 31**
CIES/9286: **Reader Service 32**

Gould offers new family of minisupercomputers

Gould has introduced the NPL family of minisupercomputers for computationally intensive engineering applications.

The new systems use Gould's UTX/32 operating system, which is a compatible multiprocessor extension of the Unix operating system consisting of AT&T System V and Berkeley BSD 4.3 environments.

According to the company, the NP1, the first series of the new family, has an open systems architecture, including parallel and high-speed vector processing, and massive memory.

Another product, the Realtime Hub, allows all Gould systems to be interconnectible and compatible. The Hub uses NP1 technology and components to permit integration of Gould's Concept/32 systems and the NP1.

The NP1 models range in price from $395,000 to $2.9 million. The largest system, the Model 480, has up to four billion bytes of physical memory. The entry-level Model 110 features up to 512M bytes of memory.

The low-end NP1 models will be available in the third quarter of 1987 and the high-end models, in the first quarter of 1988.

NP1: **Reader Service 33**
Hub: **Reader Service 34**

Wang integrates systems

Wang Laboratories has announced a system for managing, processing, and communicating text, data, and imaging information. Wang Integrated Image Systems (WIIS) combines products and services in a variety of configurations, available at prices ranging from around $100,000 for an entry-level system to $750,000 and up for high-end installations.

The company plans to ship entry-level versions in the third quarter of 1987. High-end configurations are planned for the fourth quarter of 1987.

WIIS are based on Wang's VS family of superminicomputers and on Pace, Wang's fourth-generation applicational development and relational database management system.

Entry-level and larger systems can include image workstations for full- or half-page image display, stand-alone optical drives, and scanners. Larger configurations built around a VS 65 or larger system can include cabinet-mounted optical drives, optical jukeboxes, film storage and retrieval devices, and microfilm cameras. These systems require image transfer controllers.

Reader Service 35

Unisys expands hardware and software lines

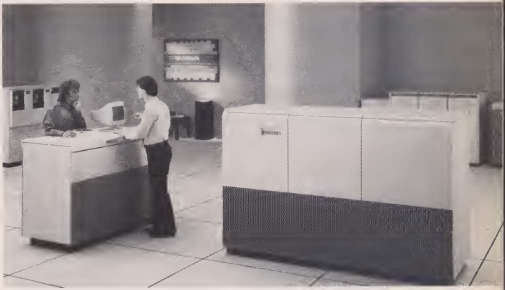
Unisys has expanded the products offered in its hardware and software families.

The company has announced two additions to the V Series of computers, the V 510 (base price $950,000) and V 530 (base price $1,775,000). According to Unisys, the models double and quadruple the performance and memory capacity of the current top-end model, the V 380. Deliveries are planned for the fourth quarter of 1987 for the V 530 and the first quarter of 1988 for the V 510.

An addition to the B 25 family of desktop computer systems, the B 38, is based on the Intel 80386 microprocessor. The system supports up to 11 cluster workstations, according to the company. Prices start at $8375.

ClusterShare allows PC workstations to coexist on the BTOS cluster with BTOS workstations. The product consists of an RS-422 interface add-on board and ClusterShare software. The card will be available in the third quarter for $475.

The InfoExec series of data management products is based on semantic data model technology.



The Unisys V 510 can be field upgraded to the V 530.

The InfoGuard software security module extends the security facilities already available on the A Series computers.

V 510: **Reader Service 36**

V 530: **Reader Service 37**
B 38: **Reader Service 38**
ClusterShare: **Reader Service 39**
InfoExec: **Reader Service 40**
InfoGuard: **Reader Service 41**

SugarCube workstation joins Intel's iPSC hypercubes

Intel Scientific Computers has announced an entry-level member of the iPSC family of concurrent computers. The SugarCube workstation reputedly broadens the hypercube-based iPSC family by offering systems designed for individual researchers, applications developers, and original equipment manufacturers.

SugarCube systems are available in four desktop-sized models: a standard system for concurrent computing research; an extended memory system for artificial intelligence and symbolic computing applications; a vector system for numeric-intensive applications; and a hybrid version that combines numeric and symbolic capabilities.

The standard SugarCube/d3 system consists of eight processing nodes for $45,950. The standard extended memory SugarCube-MX/d2 contains four processing nodes and 18M bytes of distributed memory for $49,950. The SugarCube-VX vector system consists of four nodes, each with an arithmetic accelerator, for $69,950. The four-node hybrid version ranges in price from $49,000 to $69,000, according to the customer's configuration.

SugarCube/d3: **Reader Service 42**
MX/d2: **Reader Service 43**
VX/d2: **Reader Service 44**
Hybrid: **Reader Service 45**

AI software prevents, detects, repairs hardware errors

Prime Solutions claims that their Disk Technician artificial intelligence software automatically prevents, detects, repairs, and recovers hard disk media failures before data is lost. The software runs on the IBM PC, PC-XT, PC-AT, PC-Jr., Compaq computers, and IBM PC clones.

According to the company, the program runs automatically and works on both hard and floppy disks. It reputedly performs automated daily, weekly, and monthly system testing and repairing for the ability to read and write, the soft error rate, track alignment, and magnetic retentivity. Unsafe soft errors are repaired or blocked. Programs and data files in use are moved to a safe area before data is lost.

The software also features an early warning detection system that removes unrepairable marginal areas from use before total failure. It uses factory low-level Real format to repair bad areas, and recovers disk space lost by the IBM DOS format blocking entire tracks.

Artificial intelligence reputedly helps Disk Technician adjust to the system being checked, as well as learning the host system to improve succeeding test and repair program runs.

The list price for Disk Technician is $99.95.

Reader Service 46

G-Logis is object-oriented version of Prolog

Man-Machine systems offers G-Logis, an object-oriented version of Prolog developed at Universite de Technologie de Compiegne in France and tested and enhanced by Graphael.

According to the company, G-Logis makes possible functional and procedural programming, logic programming, and object and message programming.

Features added to Prolog include no specific interpreter, integration into Lisp, dynamic clause insertion, object-oriented programming and message programming, ability to declare and use external objects, and expandability of unification and inference mechanisms.

G-Logis costs $10,000. Man-Machine Systems will market the product in the United States, Canada, and the Far East, while Graphael will market it in Europe.

Reader Service 47

Diga! for Amiga uses Doubletalk

Aegis Development has announced the telecommunications program, Diga!, for the Commodore Amiga computer.

Diga! uses the Doubletalk file transfer capability. According to the company, this means that when two computers are connected by modem, one can be sending and receiving a file while the users communicate through the keyboard.

The program requires 512K bytes of memory and one disk drive. It costs $79.95.

Diga! also reputedly lets users create custom emulations, including graphics.

Automated sessions called Scripts permit data capture, branching, and conditional statements. Scripts are stored in an address-book filing system along with settings, phone numbers, and other information about each entry.

Remote operations can be performed with password protection.

Kermit, Xmodem, and batch file transfer protocols are provided, with 10 programmable function keys available.

Reader Service 51

Workstation integrates AI

MAD Intelligent Systems has announced the RDS 3000 workstation, which the company calls a unified hardware and software system for integrating artificial intelligence technology with conventional computing.

The RDS 3000 features MAD's proprietary software, RelationalLisp. The language provides AI techniques and the ability to construct and access relational databases.

The workstation runs Unix V.3 and MS-DOS and includes X-Windows and several implementations of Common Lisp.

The hardware platform includes Intel's 80386 microprocessor and 82786 coprocessor for graphics. The RDS 3000 worksation has an AT-bus architecture, eight slots, 85M bytes of hard disk storage, and up to 16M bytes of RAM.

MAD markets its products to industry resellers, system integrators, and original equipment manufacturers.

Reader Service 48

Character-recognition tool for IBM PC line unveiled

Handprints, a new software tool from Skylight Software, includes a set of routines designed to enable programmers to incorporate character-recognition capability into their applications.

Handprints is a software implementation of the advanced on-line handprinted-character-recognition algorithm. It can be used for handprinted letters, digits, and some special characters and mathematical symbols.

A development kit for the IBM PC line includes C- and assembly-language callable routines for character entry and recognition. The kit includes an optional Mitsubishi GrafNet Model-01 professional digitizing tablet with 8-inch × 12-inch active area, 200 ppi resolution, and 60 points/sec sampling rate. Various tablets and light pens are supported.

Handprints sells for $1200, including the Mitsubishi GrafNet tablet ($395).

Reader Service 49

SQL now on Tandem NonStop computers

Tandem Computers has announced NonStop SQL software for Tandem NonStop systems. The new distributed relational database management system incorporates Structured Query Language, according to the company.

NonStop SQL is reportedly integrated with Tandem's Guardian operating system, as well as products in the company's existing database management series.

NonStop SQL will be available in the third quarter of 1987. The price varies according to the NonStop system used. The software costs $4000 for the initial license with a $500 monthly license per processor for NonStop VLX systems or $375 per month per processor for NonStop TXP and 11 systems. The initial license is $8000 with a $750 monthly fee per system for NonStop EXT25 systems, and $6000 with a $600 monthly fee per system for NonStop EXT10 systems.

Reader Service 50

Sun-3 computer is basis of imaging workstation

Vicom Systems has released Vicom-VME, a modular, deskside, image-processing workstation built around a Sun-3, VMEbus, single-board computer.

The Vicom-VME utilizes a bus structure in which image memories, dedicated processors, and the 32-bit Sun microcomputer are interconnected in a modular fashion. The Sun microcomputer and the high-speed hardware are tightly coupled on the VMEbus.

The Vicom-VME offers separate modules for image acquisition, processing, and display. By mixing and matching these modules, users can customize each workstation to meet their specific needs.

Real-time image processing needs are handled through a real-time operating system that is separate from the Unix operating system.

Pricing for a complete workstation begins at $68,000.

Reader Service 52

IBM introduces operating systems for Personal System/2 PCs

Both IBM Operating System/2 and an enhanced version of IBM PC-DOS (IBM DOS 3.3) are designed to take advantage of the capabilities of IBM's Personal System/2 PCs.

IBM's 3270 Workstation Program, when used with the 80286 Expanded Memory Adapter (XMA), will provide a growth extension to IBM DOS users who require 3270 emulation and the capability for using multiple concurrent applications.

IBM Operating System/2, which can support up to 16M bytes of memory, is intended as a full-function operating system that lets users run multiple applications while communicating with other systems. It is meant to be used with Personal System/2 Models 50, 60, and 80, as well as with IBM PC XT Model 286 and the IBM PC AT.

Standard Edition 1.0 of Operating System/2 is intended to support large memory, multitasking, and an IBM DOS operating environment. Standard Edition 1.1 of Operating System/2 will include all the functions of Standard Edition 1.0 and add Presentation Manager, a graphics-based program interface, and windowing. Standard Edition Versions 1.0 and 1.1 are designed for users who do not require communications or database support.

IBM Operating System/2, Extended Edition, will include the functions of Standard Edition 1.1 plus a relational database management system, intersystem communications, connectivity, and terminal emulation.

Standard Edition 1.0 will be available in the first quarter of 1988 for approximately $325. The availability date of Standard Edition 1.1 and the Extended Edition has yet to be determined; prices will be approximately $325 and $795, respectively.

IBM DOS 3.3 is a single-tasking operating system that can run on the entire line of IBM personal computing systems. DOS 3.3 replaces all prior levels of DOS except DOS 2.1.

New commands, such as Fastopen, Call, and Append have been added to make files within DOS easier to use. Other enhancements include additional communications ports and support of fixed disks larger than 32M bytes.

IBM DOS 3.3 can be used with other programs to provide new functions and support that are interim solutions and logical migration steps to IBM Operating System/2.

Users can upgrade to the new program for $75.

IBM 3270 Workstation Program Version 1.0, which is scheduled to be available beginning in June, will extend DOS 3.3 to allow users to get information from up to four concurrent host sessions, up to six concurrent DOS sessions, and up to two notepad sessions. It is designed for use on selected models of the IBM PC XT, PC AT, and all Personal System/2 models, as well as on IBM's 3270 PC systems.

IBM 3270 Workstation Program Version 1.1, which will be available in the first quarter of 1988, is designed to run on the Personal System/2 Model 80 and provides the functions of the XMA card without requiring the adapter. In addition, when it is used with the IBM Token Ring LAN gateway, users are provided full 3270 emulation through the IBM Token Ring without the need for the 3278/79 adapter.

Both 3270 programs will carry a price of $495. The Expanded Memory Adapter is priced at $1045 and is available now.

OS/2: **Reader Service 53**
DOS 3.3: **Reader Service 54**
3270: **Reader Service 55**
Adapter: **Reader Service 56**

Cassette-size modem weighs 3 ounces

Novation has announced the Novation Parrot 1200 modem. The 300/1200-baud modem weighs three ounces and is approximately the size of an audio cassette, according to the company. The suggested retail price is $119.

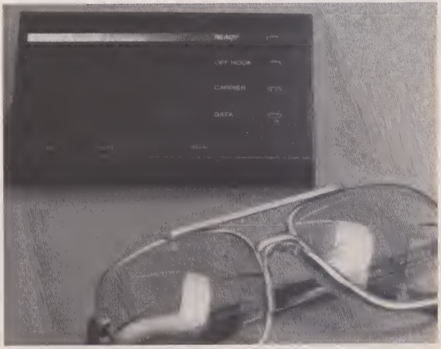
A microprocessor-controlled power management system reputedly enables the Parrot 1200 to function using the power available from the host computer's RS-232 serial interface.

The modem is compatible with Bell 103/212A modems.

Other features include asynchronous data format; full duplex operations; built-in self-test, analog loop-back, local digital loop-back, and remote digital loop-back testing; touch tone and pulse dialing; automatic answer; AT-standard (Hayes) command format; and dial tone, busy, ring back, MCI, and Sprint tone detection.

Options include adapter, cable, and software. Starter kits are available for IBM PCs and compatibles, Apple Macintosh, and Commodore computers.

Reader Service 57



Novation's Parrot 1200 modem is about the size and weight of an audio cassette.

Intel offers integrated department service network

Intel has announced a system for department automation of large numbers of personal computers.

The MultiServer system provides a common networking solution, with transparent file access regardless of where the data resides on the network. It supports personal computers, departmental level computers, and mainframes.

OpenNet MultiServer software provides a range of network services that are available to every node on the network.

At the heart of the OpenNet MultiServer network is the Intel System 300 Series microcomputer, optimized for local area network control.

The MultiServer system comes in two configurations: a high-performance base file/print server configuration; and a full file/print/communications server with eight-line communications controller for asynchronous and host-link communications. Both systems include 140M bytes of mass storage, 60M bytes of tape cartridge backup, a 320K byte floppy disk drive, an 8-MHz 80286 CPU, and an integrated OpenNet Ethernet LAN controller.

Complete MultiServer systems range in price from $15,000 to $30,000, depending on the configuration.

Reader Service 58

Kodak extends CAR product line

Eastman Kodak has announced four products that support its line of computer-assisted retrieval (CAR) information systems.

The KAR-6600, a mid-volume, standalone, computer-assisted retrieval system, can be used by six to eight persons. Based on a Prime 2350 minicomputer upgradable from 4M bytes to 8M bytes, the system is supplied with one or two disk drives for 120M bytes or 240M bytes of storage.

The KAR-6600 includes KAR information software, Version 3. With tape drive, the 240M-byte system sells for $80,300; the 120M-byte system without tape drive sells for $63,000.

The KAR mainframe system is a generic computer-assisted retrieval software package for IBM mainframes of the 43xx and 30xx families that run CICS under MVS or VSE. The package is available on license from Kodak for $30,000 with an annual renewal fee of $4500.

Kodak has also introduced the Starmate 500 universal reader-printer. The company says that the Starmate 500 has the smallest footprint of any such device on the market, and adds that it can be adapted to accept manual or motorized 16/35 mm roll film adapters. It is priced at $2700.

KAR-6600: **Reader Service 59**
Mainframe: **Reader Service 60**
Starmate 500: **Reader Service 61**

Network links 20 PCs and peripherals at $100 per node

Avatar Technologies has begun shipping Alliance ZSL, its PC network that connects 20 PCs and peripherals at $100 per node.

Alliance ZSL is designed to be a complete networking system that allows IBM personal computer users to communicate with one another and to share peripherals and other resources across the network. Those resources include printers, modems, protocol converters, and host computers.

The networking system consists of hardware, software, documentation, and cabling.

The hardware component acts as a network processor, resides at the center of the network's star configuration, and eliminates the need for a dedicated file server. With standard RS-232 serial port connections, up to 20 PCs and other devices can be located as far away as 500 feet from the controller. PCs can communicate at speeds up to 115K bps.

The Alliance ZSL software is packaged on a single non-protected diskette and includes network-management software, hot-key software, file-transfer and queued-operation capabilities, printer-server and user software, and a printer-sharing feature.

Three versions of Alliance ZSL are currently available: an 8-node version priced at $1195, a 14-node package for $1595, and the 20-node version for $1995. A user-installable 6-node upgrade kit is available for $600.

Reader Service 63

Ethernet-to-DDN gateway released to production

Advanced Computer Communications has released its ACS 4020 Ethernet-to-DDN gateway to production. The stand-alone ACS 4020 enables up to 49 host computers on the Ethernet LAN to share a single physical port on the DDN Packet-Switch Node.

Advanced Computer Communications says that the gateway is, in effect, a port expander that permits the Ethernet to appear as part of the Internet environment.

All hosts on either DDN or Ethernet that use the ACS 4020 will be using the DoD Standard TCP/IP (Transmission Control Protocol/Internet Protocol) higher level protocols. The 4020 uses ARP (Address Resolution Protocol) for dynamic mapping of Ethernet addresses onto Internet addresses.

The ACS 4020's link to the DDN network supports data rates of over 256K bps through a single physical port. Network configuration tables in the ACS 4020 can be modified and monitored from both local and remote sites.

Additional ACS 4020s can be added in parallel on the same Ethernet.

The ACS 4020 is available in a rack-mountable or a stand-alone configuration for $12,900.

Reader Service 62

The ACS 4020 provides transparent access to a single DDN packet-switch node port for 1 to 49 Ethernet hosts.

Boards support circuit design on Macintosh II, SE

Vector Electronic has announced the development of five prototyping boards and three extender boards for the Macintosh II and Macintosh SE computers.

The prototyping boards are intended to enable engineers and technicians to construct prototypes of new designs for electronic circuits and to test the prototypes prior to putting them into production. The extender boards are designed to be used to gain access to a circuit card (for test, measurement, and repair) while the board is actually plugged in and operating.

The eight products are

- Model 5085-3, the Macintosh II Pad-Per-Hole Prototyping Board ($64.50);
- Model 5085-4, the Macintosh II Voltage/Ground Prototyping Board ($69.50);
- Model 5085-5, the Macintosh II Vectorbord plus Prototyping Board ($83.50);
- Model 5086-3, the Macintosh SE Pad-Per-Hole Prototyping Board ($57.50);
- Model 5086-4, the Macintosh SE Voltage/Ground Prototyping Board ($60.50);
- Model 3690-27, the Macintosh II and SE Extender Board ($48);
- Model 3690-28, the Macintosh II High Performance Extender Board ($300); and
- Model 3690-29, the Macintosh SE High Performance Extender Board ($300).

Vector Electronic's Vectorbord plus product line utilizes a packaging technology that makes use of multilayer construction and the use of SMD coupling of alternating values.

5085-x: **Reader Service 64**
5086-x: **Reader Service 65**
3690-xx: **Reader Service 66**

Husky handheld computer weighs under 2 pounds

The Husky Hawk portable handheld computer, which has 352K bytes of RAM and 124K bytes of ROM, has been introduced by Husky Computers.

The Husky Hawk weighs less than two pounds (including 35-hour NiCad batteries); uses CMOS memory; employs the 6.144-MHz HD64B180 central processor (a superset workalike of the Z80); features a 68-key keyboard with 10 special function keys; has eight-line, 240 × 64 dot backlit LCD display; and is fully programmable, including a built-in superset version of Microsoft Basic.

The Hawk has two asynchronous RS-232/V24 ports, one a full modem interface through a 25-pin connector, the other an eight-pin mini-DIN connector.

It is priced at $1895.

Reader Service 67

Asynchronous networking system provides network management

Able Computer has introduced its Micro Integrated Network Exchange (Minx), a desktop-sized resource-management system for asynchronous switching.

Minx supports up to 480 ports in both distributed and concentrated applications.

The company says that unlike other network solutions, Minx features the combined strengths of LANs and switches: distributed networking, centralized network management, user-initiated switching, network diagnostics, security levels, port contention, load balancing, multisessions, and queuing.

It adds that Minx complements existing file-transfer networks, allowing companies to offload terminal and PC traffic to the Minx network.

Minx uses a combination of terminal, printer, and host servers to connect users and computers in a distributed environment similar to a LAN. Terminal and printer servers are connected to Minx anywhere along twisted-pair wiring—up to 2000 feet from the Minx. Long-distance connections using fiber-optic modems and T1 connections are also supported.

Minx is priced at $5000.

Reader Service 68

Minx offers diagnostic testing of network nodes, user ports, and terminals. It also initiates automatic logoff procedures.

Media-conversion system based on IBM PC-XT, PC-AT

Applied Data Communications has introduced its intelligent TM500-ST Media Conversion/Duplication System, which is designed to translate data from source diskettes to incompatible media formats without requiring additional reconstruction.

The TM500-ST offers media-conversion capabilities for more than 500 different media formats, including 8-, 5.25-, and 3.5-inch diskettes and magnetic tape.

The TM500-ST runs data through a series of translation tables that adapt existing data to a new format.

The company currently offers more than 500 conversion programs at $500 apiece.

The TM500-ST can be used with an IBM PC, PC AT, or compatible with at least 640K bytes of memory and a hard-disk drive.

The TM500-ST operates through an interface that runs on the PC. The system reads data from a source diskette, stores it on the PC's hard-disk drive, performs data conversions, and transfers data to another diskette, to data communication lines, to magnetic tape, or to any other type of media storage device.

It is list priced at $15,750 and includes a preconfigured PC-compatible system with a 20M-byte hard disk drive.

The optional RDH-525 robotic diskette handler, which has a 100-diskette hopper, allows the TM500-ST to be used for volume diskette duplication. It is priced at $6500.

TM500-ST: **Reader Service 69**
RDH-525: **Reader Service 70**

Modem for poor-quality lines

Okidata offers a 9600-bps synchronous modem for use over dial-up or dedicated telephone lines that the company claims improves performance over public lines of uncertain quality.

The CLX96DP belongs to Okidata's CLX96 Series of modems. It features automatic adaptive equalization and complies with CCITT V.29 requirements. It also meets Federal Standard 1007.

The modem operates in half-duplex mode on dial-up lines and half- or full-duplex mode on dedicated lines. It offers built-in diagnostics for fault isolation. Local analog and digital and remote digital tests provide unattended loopbacks to isolate error conditions. Remote analog testing over dedicated lines provides additional help in locating problems.

The Okidata CLX96DP modem offers 9600 bps speeds over public lines of uncertain quality.

The CLX96DP is available in stand-alone or rack-mount models. Suggested retail price is $1495.

Reader Service 71

New computers from Plexus

Plexus Computers has announced the Plexus XDP Extended Data Processing System and the Plexus P/95 departmental data processing system.

According to the company, the Plexus XDP System merges a super-microcomputer and relational database with personal computer workstations, document scanners, optical character readers, facsimile machines, laser printers, and optical disks.

The XDP System consists of two major hardware elements managed by Plexus' XDP Operating Environment software: an XDP data server and a network of XDP workstations.

The data server is based on the Plexus P/95, P/75, or P/55 systems. All are MC68020-based computers using the Unix operating system. The workstations are IBM PC-AT-compatible microcomputers with image processing capability, peripheral interfaces, and communications controllers. Workstation software executes under Microsoft Windows running MS-DOS 3.1.

Prices for the XDP range from $75,000 for a system consisting of an XDP data server, an XDP workstation, a scanner, a printer, and software, to over $1 million for a maximum configuration.

The Plexus P/95 reputedly outperforms the company's previous high-end product, the P/75. It uses the Unix operating system and is based on the 32-bit MC68020 microprocessor running at 25 MHz. It costs $57,200.

XDP: **Reader Service 72**
P/95: **Reader Service 73**

Epson enhances Equity

Epson America has enhanced the Equity I. According to the company, the IBM PC-XT-compatible Equity I+ runs at twice the speed of the XT.

The Equity I+ has a dual-speed (4.77 and 10 MHz) 8088 microprocessor, five expansion slots, 640K bytes of RAM, support for a 3.5-inch floppy disk drive, MS-DOS version 3.2, GW-Basic, and a set of system diagnostics.

The new computer comes in three configurations: single floppy disk ($1095), dual floppy disks ($1295), or single floppy with a 20M-byte internal hard disk ($1695).

Reader Service 74

Time series software

BBN Software Products has announced DataProbe software, which the company calls an interactive time series analysis and graphics system.

DataProbe contains a structured command language that permits creation of automated procedures for quick-look reports.

The software runs on Digital Equipment's VAX computers under the VMS operating system.

Prices for a DataProbe license range from $30,000 to $65,000 depending on the VAX model.

Reader Service 75

Hand-held scanner for IBM PC

Saba Technologies has announced Handscan, a hand-held data-entry device for the IBM Personal Computer.

Handscan attaches directly to IBM PCs and compatibles and works with software programs such as Lotus 1-2-3, dBase III, Rbase 5000, Wordstar, Wordperfect, and Multimate.

The information scanned is entered into an application program as if it had been typed in, according to the company. Programmable function keys and a substitution table permit formatting of data to match an application.

Handscan consists of the hand scanner, an interface board, software, and a user's manual. The company recommends 640K bytes of RAM and a hard disk. The product requires a full-length expansion slot and MS-DOS, version 2.1 or later.

Handscan's suggested retail price is $649.95.

Reader Service 76

Fairchild introduces single-chip modem family

Fairchild Semiconductor has announced a single-chip Hayes-compatible 1200-bps full-duplex modem as the first in its Surelink family of data communications products.

The uA212AT performs all standard Bell 212A/103 signal processing functions and supports auto-dial, auto-answer, and diagnostics. All analog functions are on-chip.

The chip is available in 28-lead ceramic dual-in-line packages and in plastic leadless chip carriers for surface mounting. In quantities of 1000, prices are $27.50 for the 28-pin ceramic DIP and $29 for the plastic LCC.

A design kit consisting of a Smartmodem-compatible board containing the uA212AT chip, a 6801 microcontroller, data access arrangment, electrically-programmable read-only memory, and an RS-232 interface costs $499.

Future members of the Surelink family will reputedly offer on-chip universal asynchronous receiver/transmitter, hand-shaking protocols, and mode options.

uA212AT: **Reader Service 77**
Kit: **Reader Service 78**

Comm90 aids networking

Prentice has introduced a family of data communications equipment reputedly designed to integrate the capabilities of a data PBX, switching statistical multiplexer, X.25 PAD, and LAN terminal server.

The Comm90 Series is based on a distributed architecture that according to the company provides each system module with local intelligence.

The Comm90 Series includes the DX90 Data PBX, MX90 Switching Statistical Multiplexer, PD90 X.25 Packet Assembler/Disassembler, and LT90 LAN Terminal Server. All units are compatible.

The DX90 comes in two models with capacities ranging from 16 to 336 ports. The DX90 also supports Ethernet, IBM SNA/SDLC, and X.25 interfaces. Prices start at $4600 for an entry-level 16-port unit and range to $40,000 for a fully configured system.

The MX90 with 32 input ports and two composite channels costs $7600. The 96-port unit with four composite channels sells for $18,600.

The PD90 offers up to 14 synchronous network links. It conforms to CCITT X.3, X.28, and X.29 standards. Prices start at $7600 for an entry-level 16-port unit and range to $15,600 for a fully configured system.

The LT90 is Ethernet compatible. It can attach up to 96 devices, including terminals, modems, and printers. It supports IEEE 802.3 standards and TCP/IP. No price given.

DX90: **Reader Service 79**
MX90: **Reader Service 80**
PD90: **Reader Service 81**
LT90: **Reader Service 82**

CAE software tools for IBM PC-AT

Phase Three Logic has announced the CapFast family of CAE software tools for the IBM PC-AT and compatible machines, including the Intel 80386 CPUs, with EGA or high-resolution color graphics, a hard disk drive, and 640K bytes of RAM. The basic package consists of a schematic editor, a symbol editor, a symbol library, and a plotter utility.

Optional software includes an interactive simulation grapher and netlisters.

The tools were originally engineered for CAE workstations, according to the company.

The EGA version of the basic schematic design package and a standard parts library cost $2600.

Reader Service 83

New computers out from Honeywell Bull

Honeywell Bull has introduced the DPS 7000 line of midrange 32-bit mainframe computers for transactional processing and support of integrated solutions. The family was designed and built by Compagnie des Machines Bull.

The DPS 7000 units were reportedly developed as follow-on systems to the Honeywell Bull DPS 7 line. The new line includes five models: the 10, 20, 30, 40, and 50. The units support up to 600 terminals and range in price from $127,000 to over $1 million. They run the GCOS 7 operating system.

The DPS 7000 systems will be available in the United States in August 1987.

Reader Service 84

LANcard connects IBM PCs on Z-LAN

Zenith Electronics has expanded its local area network product line with the LANcard for the IBM PC, PC-XT, PC-AT, and compatibles. The product reputedly allows personal computers to communicate at 0.5M bits per second in Zenith's Z-LAN broadband LAN.

According to the company, the Z-LAN 500C LANcard is an intelligent bus-adapter card designed for installation in a PC expansion slot. The card also has the Netbios standard network interface, which operates LAN software that supports Netbios.

The LANcard also reputedly has the built-in capability to perform layer- and system-level network management functions as well as status monitoring for simplified system installation and maintenance.

Prices range from $495 to $695, depending on volume.

Reader Service 85

TeleSoft offers Ada tools and training

TeleSoft has announced the Ada Profiler to assist developers in pinpointing areas in Ada programs where program alterations will improve performance.

According to the company, the Ada Profiler monitors the execution of programs and provides a reduced profile report of the data to help locate possible inefficiencies in applications programs.

The Profiler also includes a "what-if" feature.

The Ada Profiler is available for use with the TeleGen2 Ada development system on the VAX/VMS host and target, and VAX/VMS host to MC680X0 targets. Prices range from $1250 to $9500 according to configuration.

TeleSoft has also announced the Introduction to Ada package to augment in-house Ada training programs. The package includes an on-line document retrieval system, 50 new quizzes, and sample Ada programs. The Introduction to Ada package operates with TeleGen2 on any VAX/VMS system. Prices range from $1500 to $10,000 according to configuration.

Profiler: **Reader Service 86**
Intro: **Reader Service 87**

SmartView manages network

PCI has announced SmartView, an X.25 network management system. The software runs on an IBM PC-AT or compatible under the Xenix operating system.

SmartView accesses remote SmartNet 2000 or 3000 series processors in the X.25 network through two asynchronous ports. Once loaded, the application runs unattended. Password protection guards against unauthorized entry in the management system.

SmartView costs $1750.

Reader Service 88

NEW LITERATURE

Frost & Sullivan reports. The United Kingdom leads in creating networks in the workplace, but a new study of LANs predicts that by 1990 both West Germany and France will eclipse the U.K. in this market. See "Local Area Networks in Europe" (#E857, 307 pp., $2575) for Frost & Sullivan's theories about how and why the market will change. Frost & Sullivan, Sullivan House, 4 Grosvenor Gardens, London SWIW ODH; 01-730-3438 or Frost & Sullivan, 106 Fulton St., New York, NY 10038; (212) 233-1080.

Tab Books on expert systems, design. Michael Chadwick and John Hannah provide advice, guidance, and specific program examples on writing expert system programs for specific applications in *Expert Systems for Microcomputers—An Introduction to Artificial Intelligence* (order no. 2838, 234 pp., $14.60 paper—ISBN 0-8306-2838-X, $19.95 hardbound—ISBN 0-8306-0438-3). Tab Books, PO Box 40, Blue Ridge Summit, PA 17214; (717) 794-2191.

Designing Portable Computerized Instruments by W. R. and S. E. Penrose (order no. 2710, 262 pp., $18.60 paper—ISBN 0-8306-2710-3, $27.95 hardbound—ISBN 0-8306-1310-2) tackles the design and construction of computerized instruments specifically designed for portability.

Parallel Processing: The Cm Experience* by Edward F. Gehringer, Daniel P. Siewiorek, and Zary Segall details the insights gained by a team of researchers at Carnegie Mellon University as they built, programmed, and evaluated Cm*, an experimental 50-processor multiprocessor system. ISBN 0-932376-91-6, order no. EY-6709E-DP, hardbound, 464 pp., $40. Digital Press, Digital Equipment Corp., 12A Esquire Rd., Billerica, MA 01862. Credit card orders call 1-800-343-8321.

Video on computer design paradigm. California Newsreel has released *Computers in Context*, a video to introduce Americans to a human-resource-based computer design paradigm that has emerged in Scandinavia over the past decade. The 30-minute videocassette documents three successful applications of this "tool perspective" for designing systems that augment worker creativity. Major funding was provided by the German Marshall Fund of the United States. The video can be rented for $60 plus $5 shipping, or purchased for $290 plus $5 shipping. Contact California Newsreel, 630 Natoma St., San Francisco, CA 94103; (415) 621-6196.

Computer Industry Almanac (780 pp.) covers people, companies, products, and trends in the computer industry. It supplies facts and forecasts, with directories and rankings for companies and products. It also covers education, research, salaries, and marketing, among others. Published by Computer Industry Almanac, Inc., 8111 LBJ Freeway, Dallas, TX 75251-1313; (214) 231-8735. Distributed by Ingram Book Company in the United States. Softcover, ISBN 0-942107-01-2, $29.95; hardcover, ISBN 0-942107-00-4, $49.95.

EDI market to boom. According to "Electronic Document Interchange Markets—U.S. and International" (#732, 179 pp.) from International Resource Development, the market for electronic document interchange will double this year, from $40 million to $80 million, and pass $1 billion in 1994. IRD predicts that third-party service suppliers will be major beneficiaries of this market growth. A free table of contents and description of the $2100 report is available from IRD, 6 Prowitt St., Norwalk, CT 06855; (203) 866-7800.

New from Bowker. *Microcomputer Marketplace 1987* (ISBN 0-8352-2267-5, 1115 pp., $95) covers software, companies, publications, meetings, and contacts for organizations and individuals involved in the microcomputer industry. Entries are in alphabetical order, with a cross-indexed directory. Contact Customer Service, R.R. Bowker Co., PO Box 766, New York, NY 10011; 1-800-521-8110. For delivery outside the U.S. and Canada, contact Order Dept., Bowker U.K., PO Box 88, Borough Green, Kent, TN15-8PH, U.K.

The Software Encyclopedia 1986-1987 (ISBN 0-8352-2090-7, 1881 pp., $125) is a two-volume set of available microcomputer software. *Scientific and Technical Books and Serials in Print* (ISBN 0-8352-2263-2, 4203 pp., $159.94) is a three-volume set with entries for more than 135,000 publications. *The Directory of American Research and Technology 1987* (ISBN 0-8352-2310-8, 744 pp., $185) covers over 1500 fields of scientific research and over 11,000 companies. It includes contact information, addresses, key personnel, and descriptions of R&D activities.

New criteria from CSAB. The Computing Sciences Accreditation Board has revised its minimum standards document, "Criteria for Accrediting Programs in Computer Science in the United States." The revised criteria will be applied as practicable to programs now under evaluation, and to programs evaluated in the 1987-88 accreditation cycle running from February 1987 to June 1988. Copies are free of charge from CSAB Publications, 14th Floor, 345 E. 47th St., New York, NY 10017; (212) 705-7088.

The LAN Software Directory, second edition, covers more than 400 products in two major sections: PC LAN software and software that runs on other types of LANs. Listings are arranged alphabetically according to product name. The directory costs $137, $187 outside the U.S. Contact Architecture Technology, PO Box 24344, Minneapolis, MN 55424; (612) 935-2035.

Out of the Crisis by statistician W. Edwards Deming delineates Deming's philosophy of management: capture markets with better quality, lower prices, and increased productivity. Published by Massachusetts Institute of Technology, Center for Advanced Engineering Study. Hardbound, 507 pp., $60. Contact MIT, Room 9-234, Cambridge, MA 02139; (617) 253-7444.

Japan Computer Index '87 provides a directory in English of business hardware and software in the Japanese computer industry. Available from Infonet, 5F The 7th Industry Bldg., 1-20-14 Jinnan, Shibuya-ku, Tokyo, Japan 150; (03) 770-4483.

MICROSYSTEM ANNOUNCEMENTS

Company, Model, Function	Comments	R.S. No.
Sands Communications Lightning 24 MNP Modem	This 2400-baud modem features built-in Microcom Networking Protocol, auto dialer, battery-backed multiple number storage, a speaker, and support of both synchronous and asynchronous communications. Cost: $599.	120
C.Itoh Electronics YD-701; YD-801 3.5-inch drive; 5.5-inch drive	The YD-801 has a storage capacity of 3.2M bytes, and retains read/write compatibility with current 1M and 1.6M-byte drives; the YD-701 offers a capacity of 2.0M bytes and is compatible with 1.6M and 1.0M-byte standards. These half-height floppy disk drives use surface mount technology to resist environmental fluctuations. Cost of the YD-801: $280; YD-701, $250.	121
Benford Communications Concept Card Storage card	This secondary storage board comes in four models that utilize VLSI chip technology and stiction-free design. All models come fully formatted for DOS 3.1 and occupy 1 or 1½ slots. The CC-T20 has a capacity of 21M bytes and sells for $599; the CC-T30, 32M bytes for $699; the CC-T40, 42M bytes for $1499; and the CC-T60, 63M bytes for $1699.	122
Dataram Corp. DR-280 Add-on memory	Compatible with Data General's MV/4000, MV/10000, and S-280 systems, this 256K-byte-based memory module is available in 2M, 4M, and 8M-byte versions. It also features a memory disable switch and a user-accessible memory starting address switch. The 8M-byte version costs $12,995; the 4M-byte one, $7995; and the 2M-byte version, $4295.	123
Genoa Systems Galaxy II Tape backup	In 20M or 60M-byte cassettes or 60M or 120M-byte cartridges, this backup subsystem includes a drive, an interface card and cable, software in diskette form, and a manual-driven menu. This tape backup system can support both network connectivity (with optional software) and standalone, single-user systems. It has a backup rate of 5M bytes per minute, which can be done automatically by scheduling and presetting. The internal 20M-byte cassette is priced at $895; the external cartridge of 60M bytes is priced at $1395.	124
Fujitsu America M1928CM01 Card modem	This full-duplex, four-wired leased line modem is Trellis coded and measures 56 square inches. It features a speed of 19.2K bps, with fallback speeds of 16.8, 14.4, 12.0, and 9.6K bps, and 16-state, eight-dimensional coding. Cost: $3550.	125
Hercules InColor Card Color graphics card	This video card runs IBM-compatible PC applications for enhanced color monitors and displays 16 colors from a 64-color palette in 720- × 348-pixel resolution. Meets IBM Enhanced Color Display specifications. Cost: $499.	126
Integrated Solutions VME-ECX VME Ethernet board	This board is equipped with a LANCE processor and a Motorola MC68000 CPU at 10MHz. The 256K bytes of dual-ported RAM supports 32-bit transfers over the VME interface. Has extended addressing and can grow with the system, up to 4G bytes. Cost: $1900.	127
Matrox MG-1280 Color graphics board	Designed for the Multibus I architecture, this board includes a 32016 CPU and an HD63484 drawing processor. It can draw 1280 × 1024 images at the rate of 35,000 vectors per second and includes a 16-bit DMA port. Cost (10's): $44.95.	128
Mountain Computer Micro Bernoulli Disk drive	Internal or external drive with an average seek time of 40 ms and using 5¼ 20M-byte cartridges. Compatible with IBM, AT&T, COMPAQ, and look-alikes. Software diskette and manuals included. Cost of internal unit (with controller-adaptor): $2195; external unit (not including controller-adaptor): $2395.	129
Plessey Microsystems PME BB-1 Bubble memory	Master board contains 1M byte of memory, with up to eight 2M-byte slave storage units and performs VME transfers up to 3.6M and 2.6M bytes per second in write and read cycles, respectively. Access time is 11 ms. Cost of the master board: $3983; each slave board, $4596.	130
Tecmar Captain 286 Expansion board	This 16-bit expansion board can be expanded to 4M bytes with 256K-byte single in-line memory modules and is compatible with the EMS standard. Features an IBM-compatible parallel printer port and an RS-232 serial port. Configurable for DTE/DCE protocols and can accommodate 80286 processors running at 6, 8, or 10 MHz. Cost for 512K-byte version: $845.	131

IC ANNOUNCEMENTS

Company, Model, Function	Comments	R.S. No.
Analog Devices AD392 D/A converter	Chip includes control logic, registers, latches, and four 12-bit, high-speed, voltage-out DACs with a bus access and a settling time of 100 ns and 4 μs, respectively. Pin stake 32-lead ceramic packaging. Cost (100's): $99.	140
Bowmar/White 8090 CMOS EEPROM	Nonvolatile 1,048,576-bit CMOS EEPROM chip in a 128K-bit × 8-bit/word format with three-state output and operable from −55 °C to +200 °C. Fifty-six pin ceramic DIP packaging. Cost (100's) for high-temp.: $1322; military temp.: $1495; commercial temp.: $1295.	141
Burr-Brown DAC811JU/KU DAC	Chip includes bus-compatible 12-bit D/A converter in 0.7- × 0.29-inch package, precision voltage reference, interface logic, and double-buffered latch. Has ±¼LSB max linearity error, 4-μs settling time, and ±0.2 percent gain error. Cost (100's): $13.10.	142
Integrated Device Technology IDT75C19/29 DAC converters	This CMOS 9-bit, 125-MHz DAC converter drives a 75-ohm load to video levels with resolution of 1280 × 1024. It uses 6 × 3 segmented architecture and is packaged in 24-pin hermetic DIPs, 28-pin leadless chip carriers, and 0.300-inch THINDIPS. Version C19 features ECL-compatible inputs; C29, TTL-compatible ones. Cost (100's) : $38.50.	143
Motorola MCM61L64C, CMOS RAM	This low-power CMOS static RAM has a 30-μA data retention current (from 0-70 °C) and 45- and 55-ns access times and requires 2 V for data retention. Available in 600 mil, 28-pin, cerdip Dual-in-Line package. Cost (100's): $19.50.	144
Sony Corporation CXA1076K A/D converter	This 8-bit, flash monolithic analog-to-digital converter is rated at 200 MSPS and has an input bandwidth of 100 MHz and a power consumption of 860 mW. Packaged in a 68-contact ceramic chip carrier. Cost (100's): $385.	145
Texas Instruments TMS320C30 DSP	Featuring 60-ns execution time, this digital signal processor also has two 1K- × 32-bit single-cycle, dual-access RAM blocks, a 4K- × 32-bit single-cycle, dual-access ROM block, and a 64- × 32-bit instruction cache. Both a 32/40-bit floating-point and integer multiplier and logical ALU are included. Cost (in OEM quantities): $50.	146
Texas Instruments TMS27C292 CMOS EPROM	Pin-compatible with 2K- × 8-bit bipolar PROMs, this 2K- × 8-bit EPROM offers an access time to 35 ns and a 394-mW power dissipation at 0-70 °C. It has windowed, dual-in-line, 600-mil packaging. Cost (100's): $8.75.	147
Thomson-Motek MK41H80 CMOS SRAM	Chip contains 4K- × 4-bit RAM array and a 4-bit comparator and has an address/cycle time of 20 ns. Also features flash clear and read access. Available in 300 mil. 22-pin plastic and ceramic DIPs. Cost (140's): $15.75.	148
Toshiba America TC511000J-85/10/12; TC511001J-85/10/12; TC511002J-85/10/12 CMOS DRAMs	DRAMs are organized as 1,048,576 1-bit words and in 20-pin SO-J packages. Access time is 85, 100, and 120 ns and mode is fast page, nibble, and static column for 511000, 511001, and 511002, respectively. Cost (1000's) of 120-ns version: $23.45.	149
Toshiba America TC511000J-85/10/12; TC511001J-85/10/12; TC511002J-85/10/12 CMOS DRAMs	This CMOS dual asynchronous communications element includes a Centronics-style printer interface and has two serial I/O ports and one bidirectional parallel data port. Housed in either leaded ceramic or plastic leaded Jedec-standard 68-terminal chip carriers. Cost (2000 or more): $16.50.	150
VLSI Technology VL16C452 Interface driver	Organized as 256 words × 4 bits, this 1K CMOS SRAM has an access time of either 15, 25, or 35 ns and consumes 330 mW. Packaging is in 22-pin plastic DIPs. Cost (100's) of 15-ns version: $10.44.	151

CAREER OPPORTUNITIES

RATES: $12 per line, $120 minimum charge (up to ten lines). Average six typeset words per line, nine lines per column inch. Add $10 for box number. Send copy at least six weeks prior to month of publication to: **Marian Tibayan, Classified Advertising, COMPUTER Magazine, 10662 Los Vaqueros Circle,** Los Alamitos, CA 90720.

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, *COMPUTER* may reject any advertisement containing any of these phrases or similar ones: "...recent college grads...," "...1-4 years maximum experience...," "...up to 5 years experience...," or "...10 years maximum experience." *COMPUTER* reserves the right to append to any advertisement, without specific notice to the advertiser, "Experience ranges are suggested minimum requirements, not maximums." *COMPUTER* assumes that, since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only.

THE UNIVERSITY OF MINNESOTA-DULUTH

COMPUTER ENGINEERING. The University of Minnesota-Duluth has up to 2 tenure/tenure-track positions open 9/1/87. DUTIES: Moderate teaching load, primarily upper division in baccalaureate program. Areas of emphasis include digital system design, microprocessor-based design, applications and interfacing, microprocessor system development, digital and analog electronics, telecommunications engineer (hardware), etc. Conduct sponsored research and maintain professional activities. QUALIFICATIONS: At least 1 degree in engineering, applied physics, computer science or related field. *Assoc. professor rank*: Ph.D with minimum 6 years relevant teaching in U.S. or currently hold an associate professor rank and distinction in scholarly activity. *Asst. professor rank:* Ph.D or Master's degree with 6 years post-degree industrial hardware engineering experience. Qualifications determine type and rank of appointment. Salary/competitive. Letter of application and resume to: Dr. Nazmi Shehadeh, Dept. of Comp. Engr., UMD, 271 MWAH, 10 University Dr., Duluth, MN 55812, (218) 726-6147. Request at least 3 references to send letters of recommendation directly. Applications accepted through 6/15/87 with interviews as early as 6/1/87. The University of Minnesota is an equal opportunity educator and employer and specifically invites and encourages applications from women and minorities.

SYSTEMS ENGINEER

Design and develop software for telecommunications, including user-interface using ISAM and Faircom C-Tree. Write multi-tasking routines with VRTX, and low level protocols s/Assembly. Assist new employees and participate in company meetings. Write large customizable report packages for evaluation of telemarketing. Must have B.S./Comp. Sci. and be proficient in "C", PASCAL, Motorola Assembly and UNIX and VRTX. Salary: $31,700/yr. Job and interview in Pomona, CA. Send this ad and resume to job #MA3665, P.O. Box 9560, Sacramento, CA 95823-0560 no later than 7-10-87.

THE UNIVERSITY OF NEW HAVEN

Faculty position: The University of New Haven is seeking a qualified candidate for tenure-track appointment. Areas of greatest need are digital systems, control systems, robotics and power electronics. Ph.D. required. Industrial experience much desired. Applicants must be U.S. citizens or have permanent residency status. The University has an ideal location with ready access to urban and recreational areas. Many opportunities for challenging career in a small department. Starting date: September 1, 1987 or Jan. 1, 1988. Send resume with names of references to G.J. Kirwin, Chairman E&CE Department, UNH, 300 Orange Avenue, West Haven, CT 06516.

The University is an Affirmative Action Equal Opportunity Employer and welcomes applicants from all minority and ethnic groups.

OREGON STATE UNIVERSITY

The Department of Electrical and Computer Engineering, Oregon State University has a number of tenure-track positions available for the Fall Term 1987. The department has long maintained excellent relationships with the electronics industry within Oregon and has recently received significant program improvement funds. Foremost amongst these is a new building which will contain excellent facilities for teaching and research. In addition, the College of Engineering is developing a Computer Aided Engineering Center with considerable funding from industry and the State. The department has been targeted for special development as a Center of Excellence within the State System.

Openings exist in Computer Engineering—computer architecture, CAD/VLSI, digital communications, microprocessor applications, computer systems, etc.; and Control/Systems—robotics, digital control, digital signal processing, etc. A Ph.D. is required of all candidates, together with the ability to teach undergraduate courses in electrical and computer engineering and graduate courses in their speciality area. A demonstrated research ability and a commitment to sponsored research is essential. Rank and salary will be commensurate with qualifications and experience.

The department offers an excellent program in a most attractive area with close proximity to several major high technology corporations, including Tektronix, Hewlett-Packard, Intel, Spectra-Physics, Floating Point Systems, Mentor Graphics, etc. Interested persons should contact Professor S.J.T. Owen, Head, Department of Electrical and Computer Engineering, Oregon State University, Corvallis, OR 97331. OSU is an AA/EOE and complies with Sec. 504 of Rehabilitation Act of 1973.

CONCORDIA UNIVERSITY

The Department of Computer Science, CONCORDIA UNIVERSITY, is seeking qualified candidates for a tenure track position in the areas of Programming Languages or Artificial Intelligence. Applicants should have a Ph.D degree in computer science or related field with a good research record. The Department currently has 24 full-time professors. It offers both undergraduate and graduate programs up to the Ph.D level with an enrollment of over 1000 students. The language of instruction is English. Departmental facilities include VAX 11/750, SUN workstations, VLSI design stations, PCs, and INTEL 310s. Available university facilities include VAX 8500 and a CDC Cyber 835, plus undergraduate teaching machines. Apply giving resume and the names of at least three references to: Dr. T.D. Bui, Chairman, Department of Computer Science, CONCORDIA UNIVERSITY, 1455 de Maisonneuve Blvd. West, Montreal, Quebec H3G 1M8, CANADA.

In accordance with Canadian immigration requirements, priority will be given to Canadian citizens and permanent residents of Canada.

TECHNICAL MANAGER

Direct and manage professional staff of 6 in the design and implementation of a relational database system. Directs in house hardware system analysis and purchasing, manages the choices of peripheral lines to be stocked. Oversees trouble shooting of hardware; liaisons w/producers on repair and return. Design local area network upon customers requirements. M.S. in Computer Science required, demonstrated competence in design microprogramming of ITSIAC, ISQL and embedded SOL in PL/I, VSAM, UNIX. Fluent in Chinese. 40 hrs/wk, $3,000/mo. Must show legal right to work if employment offered. Job site: Irvine. Send this ad and a resume to Job #WS3206, P.O. Box 9560, Sacramento, CA 95823-0560 no later than 6-30-87.

UNIVERSITY OF WEST FLORIDA
The William Craig Nystul Chair in Computer Science

Nominations and applications are invited for the William Craig Nystul Chair in Computer Science at The University of West Florida. This is a $1,000,000 endowed position funded by $600,000 in private contributions and a $400,000 matching grant from the State of Florida Eminent Scholars Trust Fund. The Chair will be housed in the Department of Systems Science in the College of Business.

The Systems Science Department offers both a Business Option and a Scientific Option at the undergraduate and graduate levels. Applicants and nominees should have national stature in the academic and/or practicing computer science profession, and possession of an earned doctorate is desired. Candidates coming from academic institutions must have demonstrated teaching excellence. All candidates should have engaged in substantial scholarly research resulting in publications. Candidates should have professional experience in computer science and have actively participated in professional organizations.

The successful candidate will be expected to provide leadership and direction in teaching, research, and professional service. However, the chairholder will have wide discretion concerning how these roles are to be handled. Salary is negotiable based upon qualifications.

Nominations and applications will be accepted until the position is filled. Information should be sent to: Dr. Theodore F. Elbert, Chairman, Eminent Scholar Selection Committee, Department of Systems Science, The University of West Florida, Pensacola, FL 32514-5752.

The University of West Florida is an equal employment opportunity/affirmative action employer.

THE CATHOLIC UNIVERSITY OF AMERICA

COMPUTER SCIENCE: Applications are invited for a tenure track position at the assistant professor level starting in the 1987-88 academic year. Candidates must have a Ph.D., and a commitment to computer science education and demonstrated research ability in computer science. The candidate will play an active role in the development of the program. Send resume and three references to: Dr. Mario Casarella, Search Committee Chairman, Computer Science Program, 201 Pangborn Hall, The Catholic University of America, Washington, D.C. 20064. An Equal Opportunity/Affirmative Action Employer.

MEMBER TECHNICAL STAFF

Responsible for refining and adapting architecture of large state-of-the-art software systems currently embedded in laser printer controllers to provide the same imaging models on workstation displays for multiple non-cooperating processes and subsequently to exploit multiple processor printer controllers; responsible for design, development, and implementation of the new systems from initial prototype to final product delivery. Required: Ph.D. in Computer Science. Also required: successful prior developments in designing architecture of large software systems; developing and managing large, complex software systems; developing fault-tolerant (highly reliable) software; and developing shared memory multiprocessor hardware and software, including their operating systems. Salary: $45,000/yr. Send ad with resume to Linda Garger, Adobe Systems Incorporated, 1870 Embarcadero Road, Palo Alto, CA 94303. Must have legal right to work.

NAVAL POSTGRADUATE SCHOOL,
Monterey, CA
Position Announcement in Computer Science

The Department of Computer Science has immediate openings for faculty positions at all levels in the areas of Artificial Intelligence and Computer Architecture. An applicant should have a PhD in Computer Science or a related field and have a strong interest in both graduate teaching and research. Senior applicants must have distinguished research records. Appointments can begin at any time during the year.

The Department offers MS and PhD degrees in Computer Science supported by well-equipped instructional/research facilities and a full-time technical staff. The faculty normally teach for two quarters and conduct full-time research supported by major research programs during the other two quarters.

Please send a detailed resume and three letters of reference to: Professor Vincent Lum, Chair, Computer Science Department (Code 52), Naval Postgraduate School, Monterey, CA 93943, Telephone (408) 646-2449. NPS IS AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER.

UNIVERSITY OF WASHINGTON
Computer Engineering

Applications are invited for tenure-track faculty positions in all areas of computer engineering. Our particular interests include microprocessors and digital systems, hardware and architecture, computer aided design and engineering, networks, VLSI design and test, applied artificial intelligence, computer graphics, and signal and image processing. The Computer Engineering Program is part of the Department of Electrical Engineering which has over 40 faculty members, and 500 undergraduate and 240 graduate students. Responsibilities include undergraduate and graduate teaching, and development of strong research programs. Senior faculty positions are also available, with the possibility of an endowed professorship for candidates with exceptional qualifications. At the assistant professor level, these positions may be eligible for career development awards which provide for summer research support and conference travel for up to three years. Positions are available effective Autumn Quarter 1987 or until filled. Send resume with four references to Computer Engineering Faculty Search Committee, Department of Electrical Engineering FT-10, University of Washington, Seattle, WA 98195. The University of Washington is an EO/AAE.

SENIOR SOFTWARE DESIGNERS

Incremental Systems Corporation, a young, growing company, currently needs senior people to design and develop practical software tools for the development and maintenance of large, long-lived, continuously changing systems. Our commitment to excellence of design is accompanied by an emphasis on products which enhance programmer productivity and software reliability. The Advanced Development group functions as a close-knit team in an informal environment. Highly motivated M.S.'s and Ph.D's who have a demonstrated interest in programming language design, compiler construction, code optimization, or systems architecture; have the ability to work on a diversity of projects; and can adapt to new software technologies are strongly encouraged to respond. Knowledge of Ada® is an additional plus, as is an interest in expert systems or human factors. Send your resume to: Incremental Systems Corporation, 319 S. Craig Street, Pittsburgh, PA 15213.

NEW MEXICO STATE UNIVERSITY
AI Software Specialist
Computing Research Laboratory

The Knowledge Systems Group at CRL is looking for an AI Software Specialist to support a funded project in the application of knowledge engineering to atmospheric science. The successful candidate will design and implement a prototype system, using a Symbolics 3650 color system and Common Lisp, and participate in theoretical work in the area. Experience of a Lisp Machine environment and knowledge systems programming is desirable.

In addition, the Group is seeking a Research Assistant to undertake research in support of the Group's activities in knowledge-based data interpretation. The candidate for this post will be expected to pursue a PhD in the Computer Science Department.

The salary for the AI Software Specialist is competitive and commensurate with experience. Both positions are open for immediate appointment, and American citizenship is desirable, but not essential.

Applicants should send a resume and names of three referees to: Dr. M.J. Coombs, Knowledge Systems Group, CRL, Box 30001, New Mexico State University, Las Cruces, NM 88003-0001. Deadline for applications June 31, 1987.

N.M.S.U. IS AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER.

NORWICH UNIVERSITY

Faculty position in Computer Science Engineering. Tenure-track position, rank and salary commensurate with experience. Ph.D. desired, but will consider Masters with experience. Available 1 July, 1987. Recently implemented Computer Science Engineering program. Facilities include: VAX Cluster, MicroVax supermicrocomputers and PC based workstations. New faculty will assist in laboratory and curriculum development. Expertise in any of the following areas desired: Digital Systems Design, CAD/CAE, Computer Architecture, Computer Communications and Networking. A strong commitment to undergraduate education is essential. Norwich University is located in an area of Central Vermont that offers small town or rural living with good schools and outstanding recreational opportunities. U.S. Citizen or Permanent Resident preferred. Send resume and references to Dr. Michael C. Murphy, Head, Computer Science Engineering Department, Norwich University, Northfield, VT 05663. Norwich University is an equal opportunity employer.

POST DOCTORAL RESEARCH POSITION

Electrical Engineering Department, University of Southern California. Temporary full time position for one to three years commencing immediately, with possibility of extension contingent upon funding. Salary $44,000 to $50,000 per annum, dependent on qualifications. Assist in development of knowledge based system for design of testable digital systems, with professors and graduate students. Ph.D. degree or equivalent required, along with some research experience in VLSI design, testing and design for test, and a strong background in A.I. and software engineering. Send Resume to Professor M.A. Breuer, Electrical Engineering-Systems Dept. SAL 324, University of Southern California, Los Angeles, CA 90089-0781. USC is an equal opportunity, affirmative action employer.

TEMPLE UNIVERSITY
Faculty Positions
Department of Computer and Information Sciences

The Department of Computer and Information Sciences has tenure-track and adjunct faculty positions available in the area of Information Systems.

Applicants should hold the Ph.D degree in Information Systems, Computer Science, or a closely related field. The ability to contribute to strong instructional and research programs, both graduate and undergraduate, will be the primary requisite for appointment. Salary and rank will be determined by the appointee's experience. An applicant for a senior position should have a strong record of scholarly achievement, while an applicant for an assistant professorship should present evidence of research potential.

The Department of Computer and Information Sciences offers programs leading to the Bachelors, Masters and Ph.D degrees in Business Administration/Information Science as well as in Computer Science. Currently the Department has a full-time faculty of 27, 300 graduate students enrolled in the masters and Ph.D programs, and 1200 undergraduate majors. A networked departmental computing facility includes a VAX-11/780, VAX-11/750, and 20 microVAX systems configured as time-shared systems and workstations. The facility also includes an AT&T 3B2, TI Explorer, and an image processing laboratory. Departmental instructional laboratories use VAX (VMS or UNIX) and PDP-11 systems, as well as microcomputers. University facilities currently in use by the department include a CDC Cyber 750, an IBM 3081, an IBM 4381 and several microcomputer laboratories. A multi-processor, high speed computing facility is also available for use.

Temple University is an Equal Opportunity/Affirmative Action Employer and specifically invites and encourages applications from women and minorities. To apply, submit vitae, and bibliography to:

Eugene Kwatny,
Chairman of Faculty Search Committee,
Department of Computer and Information Sciences,
Temple University,
Computer Activities Building, 38-24,
Philadelphia, PA 19122.

SYSTEM ENGINEER

Responsible for performing feasibility/engineering studies for computer integrated manufacturing projects; working with customer to define requirements of systems; performing functional and software design; leading managing teams to design and implement processes and computer hardware systems and software; managing subcontractors; and project-managing to achieve schedule and cost goals. B.S.E.E. or C.S. 4 years experience in systems engineering. Must have background in computer science. Experience in real-time software design and implementation. Knowledge of electronic assembly and automotive manufacturing processes, electrical and mechanical devices for automation, IBM-PCs, DOS, HP 1000, RTE OS, PASCAL and C. Experience in applications engineering, defining customer requirements and project management. Knowledge of manufacturing management control systems. Salary: $33,000/year. Job Site: Palo Alto. Send this ad and your resume to #SK 3083, P.O. Box 9560, Sacramento, CA 95823 not later than July 1. EOE.

UNIVERSITY OF KARLSRUHE
Federal Republic of Germany

The Department of Computer Science has an opening for the tenured position of a

Professor (C4)

for

Technical Computer Science

Successful candidates will have demonstrated research potential in at least one of the following areas

- development of algorithms for VLSI
- modelling of VLSI circuits
- structural optimisation for VLSI and WSI
- methods of verification and simulation

Candidates are expected to share teaching responsibilities for introductory Computer Science courses and advanced courses in Technical Computer Science. They should also be prepared to contribute to an advanced teaching programme in the area of VLSI design. The department is one of the well-known schools and research institutions of Computer Science in Germany. It offers a complete programme in computer science, in close cooperation with national and international establishments of advanced technology. A wide spectrum of powerful computing facilities including latest mainframe architectures as well as workstation environments is available to support experimental research and teaching.

Applicants who fulfil the qualifications for taking a professorial position should submit a curriculum vitae, a resume of experiences and interests and a list of publications not later than 31, May 1987 to the Dekan der Fakult8at f8ur Informatik, Universit8at Karlsruhe, Postfach 6980, 7500 Karlsruhe, Federal Republic of Germany.

PARALLEL PROCESSING RESEARCH ENGINEER

Utilize education and experience in research, development, and implementation in the areas of electrical engineering, computer science, and software and hardware development, including the design and implementation of at least one computer architecture for symbolic processing, and proven ability to write computer programs and build hardware systems, to perform advanced research and development in the fields of parallel processing, symbol manipulation, software technology, computer architectures, and development of end-user application programs primarily in the areas of artificial intelligence, image processing, vision, expert systems, and geographic databases. Perform advanced research for the specific development of application abstractions, computer languages (especially for parallel processing), parallel execution models, and novel architectures/organizations involving multiple computers. Utilize proven management and supervisory skills to direct other researchers both individually and in teams in performing research in the above areas. Utilize proven communications skills to communicate results of research. Applicants must have a Ph.D. in Computer Science as well as 10 years' experience in university teaching and/or proprietary research. 40 hours/week; $70,000/year. Apply at the Texas Employment Commission, Austin, Texas, or send resume to the Texas Employment Commission, TEC Building, Austin, Texas 78778, J.O. #4793110. Ad Paid by an Equal Employment Opportunity Employer.

Innovation

EMBEDDED COMPUTER SYSTEMS ANALYST

Lockheed Missiles & Space Company has an immediate opening for a computer systems analyst with five or more years' experience in the design and implementation of computer architectures and software for embedded applications.

Your applications experience should include:

- Remote Sensing
- Image Processing
- Correlation and Tracking
- Attitude Control
- Kalman Filtering
- High Reliability Computing

The successful candidate will have a BS/MS in EE, Physics, Computer Science or Applied Math. Experience with parallel processing, high processor reliability and fault tolerance is desired.

For immediate consideration, please send your resume to Lockheed Missiles & Space Company, Professional Staffing, Dept. 530GWDI, P.O. Box 3504, Sunnyvale, CA 94088-3504. We are an equal opportunity, affirmative action employer. U.S. citizenship is required.

Lockheed Missiles & Space Company
Giving shape to imagination.

“YOU CAN’T DO THAT”

Build a large scale mainframe computer that will outperform the competition’s leading model?

“IMPOSSIBLE!” they said.

But Amdahl did it back in the early 1970’s. And today we are a leader in the development, manufacturing, marketing and support of general purpose and scientific computer systems, storage products, communications systems and software.

In less than two decades we have grown from 5 to more than 7,000 “can do” employees around the globe. Our success is a result of teamwork, innovation and commitment to achieve the impossible. If you are ready for challenge, creativity and growth, explore your opportunities with Amdahl in one of the following developmental areas.

OPERATING SYSTEMS

YOU CAN develop a new, large mainframe operating system...self-tuning and does not require system generation. Use your 3+ years’ MVS and/or VM operating system development experience and your 370 Assembler language programming skills to join us in one of the following key areas:

- **SECURE SYSTEMS**
- **OPERATING SYSTEM DEVELOPMENT**
- **TRANSACTION PROCESSING**
- **NETWORKING**

YOU CAN apply for these positions by **calling CHARLOTTE MUDD at (800) 538-8460, extension 8648.**

COMMUNICATIONS

YOU CAN join an entirely new project to bring program compatibility and interoperability to the SNA and OSI environments. We’re using some of the most sophisticated software tools to do it. UNIX* System V Release 3 STREAMS for one. Our own mainframe UNIX* timesharing system — UTS** — for another. Use your 3+ years’ SNA, OSI or X.25 development experience in the following positions.

- **NETWORK PROTOCOL ARCHITECTS**
- **NETWORK MANAGEMENT ARCHITECTS**
- **COMMUNICATIONS SOFTWARE DESIGNERS & DEVELOPERS**
- **OPERATING SYSTEM DEVELOPERS & SUPPORT ENGINEERS**

YOU CAN apply for these positions by **calling ZIG SHUSTER at (800) 538-8460, extension 3285.**

UNIX*

YOU CAN “make” the kernel in 3 minutes! Use your 3+ years’ development experience in UNIX* and/or 370 operating systems to join our UTS** development team in the following positions:

- **OPERATING SYSTEM DEVELOPERS**
- **OPERATING SYSTEM VALIDATION ENGINEERS**
- **TOOLS DEVELOPERS**
- **NETWORK SOFTWARE DEVELOPERS**

YOU CAN apply for these positions by **calling BILL McCARTHY at (800) 538-8460, extension 8843.**

YOU CAN enjoy the benefits and competitive salary you would expect from an industry leader. To apply, call the designated representative for your area of interest, or send your resume to the designated representative at: **Amdahl Corporation, Employment Department, Dept. 6-2, P.O. Box 3470, M/S 300, Sunnyvale, California 94088-3470.** Principals only please.

YOU CAN AT
amdahl

---

*UNIX is a registered trademark of AT & T
**UTS is a trademark of Amdahl Corporation

Amdahl Corporation is proud to be an equal opportunity employer through affirmative action

BOOK REVIEWS

Editor: Wiley McKinzie, School of Computer Science and Technology, Rochester Institute of Technology, Rochester, NY 14623; Compmail, w.mckinzie; CSnet, wrm@rit

Message from the Editor

Spring is the season of new beginnings, and so it is with the Book Reviews department of Computer. *In May, I assumed the editorship of this department. By way of introduction, I am Wiley McKinzie, director and professor of the School of Computer Science and Technology at Rochester Institute of Technology and past editor of the Book Reviews department of* IEEE Software.

Book Reviews will continue to be the readers' department. That is, the reviews will come primarily from Computer *readership contributions. I expect to publish four to six reviews in each issue. The selection of books reviewed will lean toward the hardware and system software areas, but will include more on software engineering (the strongest interest of* Computer *readers surveyed in 1985). Other topic areas will include PCs, mini/microcomputers, engineering applications (e.g., CAD/CAM and robotics), computer graphics, database, and AI.*

There will also be something new. Two or three issues of Book Reviews a year will be devoted to special topics. Usually, the special topic will be an alternative technical knowledge source such as public information retrieval services or teleconferencing. Occasionally, the special topic will cover a humanist areas, such as computer history, social issues, or law.

Books, like politics and religion, can be discussed by anyone who has an interest in them. (Unlike politics and religion, however, a discourse about books is usually intelligent and informative.) Therefore I look forward, as editor of the Computer *Book Reviews department, to moderating this forum for the readers who love books and love to discuss them.*

—Wiley McKinzie

Operating Systems: Structures and Mechanisms

Philippe A. Janson (Academic Press, New York, 1985, 267 pp., $32.50)

Intended as supplementary material for an elementary course in operating systems, this book does accomplish its main objective. *Operating Systems* emphasizes the organization of operating systems and describes various mechanisms that are inherent in them. It deals almost exclusively with timesharing systems, although single-user systems are mentioned at times to contrast differences.

Janson assumes a basic background in computer architecture and programming languages and techniques. Also, some hardware knowledge is required to understand the concepts of system interaction. One skill the reader does not need is a degree in probability and calculus, as is needed in using some other operating systems books. What theory this book does explain isn't steeped in arcane mathematics. This nonmathematical approach makes the text an excellent source of information for someone new to the topic.

In Section 1 Janson introduces the reader to the basic functions of operating systems. Discussing such topics as resource management and virtual machines, he quickly moves to the most interesting function of the operating system, the layering of software. He does a good job of developing the concept of layers, such as the kernel and the supervisor. He also does a good job of describing the differences between batch and demand modes.

In Section 2 Janson jumps into the heart of operating systems by discussing processor management. He talks about such basic concepts as mutual exclusion, serialization, synchronization, and deadlocks. He also acquaints the user with the basic fundamentals of multitasking, task scheduling, and multiplexing algorithms. The contents of this section alone could constitute a book. I feel more time could have been devoted to discussing some more-used algorithms without compromising the book's broad approach. This section also covers the topic of task switching, an area of great concern to those trying to understand the fundamentals of microprocessors and how they integrate nicely into multitasking environment. Overall, I found this section to be the most interesting part of the book.

Section 3 discusses I/O and device control, device sharing, synchronization, buffering, and scheduling algorithms. This is also very interesting but brief reading. One topic missing from this section is the concept of a device driver, or the actual software driving a specific device. This is a fundamental function of peripheral devices and should have been covered.

Section 4 covers memory management. It first discusses addressing schemes, such as absolute and relocatable, then leads into swapping and paging. Following this is the discussion of virtual memory, and associated algorithms for implementation of virtual memory schemes. Janson does a good job explaining many of the problems associated with memory and its utilization.

Section 5 studies high-level mechanisms. Starting with file management mechanisms, it presents catalog structures, file access, file backup, and data integrity. It then moves to addressing and naming, discussing linkers and the method operating systems use to interpret names and translate them to addresses. These concepts enlighten the reader as to how the operating system actually can do such functions as overlaid tasks and shared instruction and data code. The last high-level mechanism studied deals with protection and security. Included are various protection schemes. Such schemes as encryption, access control, and passwords are followed by peripheral and central protection. This section is indeed an important yet unacknowledged part of the operating system.

Recently published books and new periodicals may be submitted for review to editor at the address given above.

Note: Publications reviewed in this section are not available from the IEEE Computer Society; they must be ordered directly from the publisher.

Section 6 concludes with guidelines for system design. It discusses the system interface, system calls, and command languages. The user interface, or what the user actually enters and gets back in response, is perhaps one of the most important parts of the operating system and, in many cases, the most poorly designed.

All chapters have a section following them entitled "Suggested Reading and Classics," in which the author has broken down the main topics discussed and recommends further reading on particular subjects. In this section alone there is a wealth of information. In addition, in some chapters there is a second section entitled "Further Reading and References," which contain other books on the topic covered in the previous chapter. The book also contains an index and a miniglossary.

Overall, Janson succeeded in his goal of producing material for introducing students to the organization and structure of operating systems, while preparing them for more in-depth studies into particular related topics. The author's writing style is direct and lucid. One weak point was the book's approach to illustration. While most figures are clear and easy to understand, some of the more complicated figures are very confusing, and the lack of explanation left me wondering what the author was trying to say.

Because of the book's broad approach, I would recommend it to anyone wanting an overall view of operating systems and, perhaps, a reference book for books on this topic.

Jeff S. Ebert
Unisys

Spectral Techniques and Fault Detection

M. G. Karpovsky (ed.) (Academic Press, Orlando, Fla., 1986, 608 pp., $44.50)

This book contains revised and expanded versions of papers presented at the International Workshop on Spectral Techniques and Fault Detection, held in Boston in October of 1983. It starts with a 33-page introductory chapter providing some of the needed mathematical background, as well as an overview of the rest of the book, and ends with a 71-page appendix containing a 618-entry bibliography on spectral techniques. Both of these have been contributed by the book's editor.

The rest of the book is divided into nine chapters, as follows:

(1) Synthesis of Encoded PLAs, by R. J. Lechner and A. Moezzi, 56 pp.;

(2) Spectral Processing of Switching Functions Using Signal-Flow Transformations, by P. W. Besslich, 51 pp.;

(3) The Chrestenson Transform in Pattern Analysis, by C. Moraga and K. Seseke, 35 pp.;

(4) Filtering in a Communication Channel by Fourier Transforms over Finite Groups, by E. A. Trachtenberg and M. G. Karpovski, 38 pp.;

(5) 3D Cellular Arrays for Parallel Cascade Image/Signal Processing, by M. J. Corinthios, 82 pp.;

(6) Universal Testing of Computer Hardware, by M. G. Karpovsky and L. B. Levitin, 71 pp.;

(7) Spectral Techniques for Fault Detection in Combinational Circuits, by D. M. Miller and J. C. Muzio, 49 pp.;

(8) Signature Techniques in Fault Detection and Location, by S. J. Upadhyaya and K. K. Saluja, 56 pp.; and

(9) The Design and Analysis of High-Speed Logic Systems with Distributed Error Correction, by G. R. Redinbo, 55 pp.

The book is technically sound. Authors of most chapters are well-known authorities in their respective fields. The authors' styles of presentation are also quite good. In Chapters 6 through 10, which I checked more carefully than the rest of the book, no technical slip and a few typos and stylistic errors (mostly in Chapter 6) were noticed. I am particularly impressed by the thoroughness and accuracy of the material in Chapter 8, although Chapters 7 and 9 are also quite well-written. As a whole, the book can be considered a welcome addition to the literature in the field of computer engineering.

The physical appearance of chapters range from beautifully typeset and illustrated (Chapter 5) to typewritten with some poorly aligned hand and transfer-sheet lettering (Chapter 6). The use of different typefaces, point sizes, pitches, line spacings, margins, and formats (e.g., for references) has made the book aesthetically displeasing. My other complaints include long lists of unsorted references at the end of most chapters, the appearance of single-line "widows" at the top or bottom of some pages, and the two-year publication lag despite the use of "Rapid Manuscript Reproduction," as announced on the copyright page. On the positive side, I should mention the subject index, which is a rarity for conference proceedings. Inclusion of a name (author) index would have been helpful in view of the separate end-of-chapter bibliographies.

This is definitely a library book. I, for one, would have elected to reproduce the two chapters that are of direct interest to me rather than to purchase a personal copy of the book. The bibliography alone makes this book a worthwhile addition to computer science research libraries.

Behrooz Parhami
University of Waterloo

A Workbook for Software Entrepreneurs

A. L. Frank (Prentice-Hall, Englewood Cliffs, N. J., 1985, 118 pp., $30)

In his preface the author describes the book as a stand-alone publication providing extensive "how to" advice for the novice and a checklist mechanism for the experienced. After going through the book, my own opinion is that it is more a checklist than useful advice. Most of the topics are too sketchily treated to be of any real value, and the author seems to have attacked the task of writing the book as just another entrepreneurial venture. I don't think the 16 skimpy chapters are worth the cover price, except for a few exceptions, such as the section on brainstorming.

Most of the sections consist solely of a set of questions, with the answers going abegging. For example, Q13.5 is: "Do you utilize a team selling approach?" The ramifications of an affirmative or negative answer are not explored.

Some highlights:

Chapter 1 deals with basic concepts and discusses the types of entrepreneurs. Chapter 3 describes different forms of businesses like corporations and partnerships. Chapter 5 is headed Product Definition and discusses the formulation of product ideas, market surveys, and the uses of brainstorming. Chapters 10 through 13 are devoted to sales, and cover such topics as sales pitch, product knowledge, presentation skills, follow-up, and feedback. Chapter 16 takes the cake for brevity. It is one page long and consists of 13—yes, you guessed it—questions. On that unlucky note the author ends the book, and it will be a canny entrepreneur indeed who can benefit from this volume.

Chandan Sen

Systems Software Tools

Ted J. Biggerstaff (Prentice-Hall, Englewood Cliffs, N.J., 1986, 317 pp., $23.25)

This book demands much of its readers in exchange for an introduction to the intricacies of systems programming.

Rather than develop either a miniature version of an extant operating system or an artificial operating system for a nonexistent machine, Biggerstaff has chosen to develop tools that extend the capabilities of MS-DOS. The book cannot be used therefore as a textbook in a standard operating systems course because it covers only those aspects of operating systems needed to understand the projects discussed therein.

Two projects are chosen as illustrations of the richness of systems programming: a terminal emulator and a multitasking window system. Since *Systems Software Tools* is intended to be a "stand-alone" book that should appeal to a diverse readership, ranging from hobby enthusiasts and advanced undergraduate and masters students in computer science to practicing software engineers, the author provides background information that will make the book accessible to all readers. To this end, the book begins with summaries of both the C programming language and MS-DOS BIOS functions. While the summaries are excellent, neither provides sufficient information to understand the projects that follow. A standard C reference such as *The C Programming Language* (Kernighan and Ritchie, Prentice-Hall, 1978) is crucial to appreciate the projects fully. Needless to say, the C programming experience is also desirable. Readers must also arm themselves, Biggerstaff indicates, with the *IBM Technical Reference Manual* (Boca Raton, 1983) and the *IBM Disk Operating System Manual* (Boca Raton, 1983). I think that an IBM PC assembly language manual, for example, *Assembly Language Primer for the IBM PC and XT* (Lafore, Plume/Waite, 1984) would also be useful.

With the requisite manuals I prepared to try my hand at systems programming. I began by trying to write some of the simple screen manipulation functions, for example, scrolling the window, moving the cursor, and writing a letter on the CRT, which would be utilized later by the multitasking window system. All of these routines take advantage of a library function "sysint" offered by the C86 compiler that permits direct access to the 8086/88 "int" instruction. Much to my amazement several of the functions did not perform correctly even though a lengthy perusal of the references indicated that Biggerstaff's code was correct. One-hundred twenty pages later the source of my problem was revealed: the functions were designed for a monochrome monitor, and the C86 "sysint" function obviously did not appreciate my color monitor. In fact, only in the middle of the book does Biggerstaff indicate the hardware and software required to utilize the book fully. The list includes an IBM PC with a monochrome monitor and *at least* 256K memory, preferably with a hard disk. In addition, a C compiler that fully implements the Kernighan and Ritchie standard and an 8086/88 assembler are required. Since the projects rely on several nonstandard library functions developed by Computer Innovations for its C86 compiler, this is obviously the compiler of choice. This information ought to have been stated in the introduction.

The easier of the two projects is a terminal emulator, and Biggerstaff's discussion of the implementation of this piece of software is thorough. A detailed analysis of the 8259A interrupt handler and the 8250 UART is provided. Complete and highly readable summaries of each of the major functions of the emulator are included. In addition, the C code used in the implementation is both elegant and well-documented.

The discussion of the multitasking window system is less satisfactory. In part, the difficulties with this discussion reflect the greater complexity of the task. The project requires creation of an independent real-time process scheduler to handle situations that MS-DOS was not designed to do, for example, multitasking. Thus, several MS-DOS interrupt functions must be either rewritten or redirected. Such redirection requires a lower-level implementation than C is able to provide; hence large portions of code are written in 8086/88 assembly language. Because the discussion is spread out over three chapters designed to cover independently the major components of the system, that is, basic data structures, process management, and the windowing system, the reader is constantly flipping pages to find the interfaces between the components. This format makes it very difficult to arrange the components in a coherent whole.

In addition, the intermixing of discussions of C routines and 8086/88 routines does not improve the text's readability. Unlike the terminal emulator, the implementation of the multitasking window system is not complete. Several required routines, most notably the queue manipulation functions, must be implemented by the user. There is also a heavy reliance on C86 nonstandard library routines.

In short, readers will have to have a thorough knowledge of C, 8086/88 assembly language, and the structure of MS-DOS to benefit from *Systems Software Tools.* Readers without such a background will still experience the pleasure of creating a usable software tool, but they will be unable to modify the projects created by Biggerstaff or to program the extensions specified in the chapter exercises.

In spite of the difficulties in using the book, Biggerstaff is to be commended for trying to write, to my knowledge, the first system software introduction designed to appeal to a wider audience.

Susan Anderson-Freed
Illinois Wesleyan University

Resilient Computing Systems: Volume I

T. Anderson (ed.) (Wiley & Sons, New York, 1985, 250 pp., $34.95)

The target audience for this book is not entirely clear. Some chapters presume the reader to have great depth of understanding of the subject. Others—those devoted to philosophic discussion or system description—presume little foreknowledge on the reader's part, though they do assume reasonable background in the computer field. The book's best market is probably the high-level systems analyst or designer, with newly acquired interests in working with resilient systems. Some software or hardware specialists will also find the book offers added perspective on the total environment in which they may be operating.

The editor is British and contributing authors are from several countries. Some readers will have to accommodate to occasional British phraseology and spelling (original publication was in London). The book contains very little overlapping material, in contrast to the pure compendium of already published papers. On the other hand, there are major gaps, such as when related topic areas are noted by some of the authors but never expanded elsewhere.

About the first two-thirds of the book deals with the methodology of resilience—the remainder discusses real-life applications. The editor's sequence is fairly rational, but diving into the early chapters may turn off all but the most determined academics. Most readers should skim the introductory Chapter 1, Fault Tolerant Computing, then jump to Chapter 9, Commercial Resilient Systems, an excellent introductory treatise, and continue to the end of the book. This will provide a perspective of reality before tackling the more esoteric chapters. The remaining chapters (2 through 8) may be read (studied) in any sequence—there is little interdependence. But you may want to save Chapter 2 for last—it will take as long as the rest of the book.

The author enjoins in the Preface: "Thus if you are an implementor or a user of a computing system for which frequent and regular breakdowns are of negligible consequence, you may safely ignore this volume. Others do so at their own risk." This is an empty threat. This book will make neither the novice nor the established computer professional expert on resilient systems. It *will* make one aware of how complex the subject is and how naive we still are in the *entire* computer community (in spite of the wealth of published work) and provide inklings of the very high price tags for truly resilient systems.

A serious shortcoming in the book is the lack of extended treatment of the systems aspects of providing resiliency. Chapter 3, Design Fault Tolerance, cites practical uses of some form of replicated software with comparators, "... in systems for railway signaling, nuclear reactor shutdown, and in the 'space shuttle'—the NASA Space Transportation System." Coincidentally, and regrettably, subsequent to the book's first publication (1985), an implementation of each of these systems has suffered a major disaster. These events, though not attributable to the associated real-time computing systems, highlight the vital nature of *total* system consideration when it is the total task, process, or mission that is at issue. There are no tidy answers. Each disaster came down to the errors made by people—but they were all people who, in some sense, functioned as part of the total system. System considerations are alluded to by several of the chapter authors, but none, including the author/editor, perceived them as within the scope of their subject.

A related deficiency is the casual treatment of before-the-fact analysis of failure mechanisms, their probability of occurrence, and their consequences. Methodologies with names like failure mode and effect analysis, fault tree analysis, and similar phrases are quite well developed. Their extensive application across the total system (including people) is essential in architecting any resilient system. This is the only way to determine where the emphasis—and money—should be spent to optimize the real-life system behavior with respect to its resiliency objectives. The treatment of fault tree analysis in the chapter on software safety only muddies the water.

A serious shortcoming in the book is the lack of extended treatment of the systems aspects of providing resiliency.

Chapter 2, Hardware Fault Tolerance, is very comprehensive in its coverage, including considerable history and a 10-page bibliography—but it is not easy reading. The reader inexperienced in this field will spend many hours researching the references to understand the extensive (and essentially unavoidable) jargon. This is a necessary compromise in packing a reasonably comprehensive treatment into a single chapter.

Software Safety, Chapter 7, leaves much to be desired. The subject really deals with computers strategically used in safety-critical systems where failure could produce extraordinary losses (e.g., death). In such systems the only realistic approach is a totally integrated system strategy. The author's frustration in trying to isolate *software* safety as a separate issue seems apparent. The attempt to demonstrate the application of fault tree analysis to software is quite ineffective. The associated diagrams only confuse the matter further. Strangely, safety-critical *hardware* is not explicitly dealt with anywhere in the book.

Software Reliability Prediction, Chapter 8, presents some classical material. It then concludes with a good discussion of factors underlying the ongoing paradox: software continues to grow in importance in overall reliability/resiliency considerations, yet our prediction capability, despite the mass of publications on the subject, continues at a level of almost trivial competence.

The importance of software reliability is mentioned throughout the book, but it is put in perspective only in the system description chapters. The real-life system designers must face up to the need for some software nucleus of "absolute" integrity that can start (or restart) the system and at least handle a rudimentary set of operating system tasks before more elegant redundancy and voting schemes can be set in place. (In some systems this presumed-reliable core is actually quite extensive, residing in the mainstream decision making path for redeployment of system resources in failure situations.) I would like to see at least a chapter devoted to the realities of creating such super-reliable software nuclei.

The "methodology" chapters not mentioned above include Reliable Communications (4), Resilient Real-time Systems (5), and Robust Distributed Programs (6). Each offers pertinent material, but the treatment is moderately superficial for various justifiable reasons. "Communications" has been the subject of many books—this philosophical treatment, using a layered approach as a reference model, exhibits the broad pertinent issues but can't really deal with them in a single chapter. The "real-time" chapter offers an adequate philosophical view. However, most of the issues are covered more meaningfully (and understandably) in Chapter 13, August Systems Industrial Control Computers. The analytical treatment of programs for "distributed" systems is rather cryptic. More comprehensible (though less theoretical or broadly applicable) perspectives are offered in the networking discussion of Chapter 10, The Tandem Non-stop System.

The other "descriptive" chapters describe the tandem nonstop system (10), the MOMENTUM high-resilience system (11), the STRATUS computer system (12), and August Systems industrial control computers (13). All are well done, with well-chosen subsets of pertinent detail to convey the ideas appropriate to this text.

Overall, the book is a useful but not exceptional contribution to the literature on the rather specialized area of resilient systems. The extensive bibliographies in several of the chapters are valuable entry points to the literature of this field.

Charles B. Stott

Computer Science from Academic Press . . .

Journal of Complexity

EDITOR **J. F. Traub** *Princeton University, Princeton, New Jersey*

Publishes original research containing mathematical results on complexity, focusing on problems approximately solved and for which optimal algorithms are available.

Volume 3 (1987), 4 issues **ISSN 0885-064X**

In the U.S.A. and Canada: $80.00 **All other countries: $93.00**

Journal of Parallel and Distributed Computing

EDITORS-IN-CHIEF

Kai Hwang

Departments of Electrical Engineering and Computer Science, University of Southern California, Los Angeles

Leonard Uhr *Computer Sciences Department, University of Wisconsin, Madison*

Directed to researchers, engineers, educators, and users with interests in parallel processing and/or distributed computing.

Volume 4 (1987), 6 issues **ISSN 0734-7315**

In the U.S.A. and Canada: $119.00 **All other countries: $148.00**

Information and Computation

(formerly *Information and Control*)

EDITOR **Albert R. Meyer** *Massachusetts Institute of Technology, Cambridge*

Publishes articles in all areas of theoretical computer science, information theory, and control theory, with emphasis on articles reflecting computational or interdisciplinary concerns.

Volumes 72–75 (1987), 12 issues **ISSN 0890-5401**

In the U.S.A. and Canada: $368.00 **All other countries: $427.00**

Journal of Computer and System Sciences

MANAGING EDITOR **E. K. Blum** *University of Southern California, Los Angeles*

Focuses on the publication of research in computer science and in system science, with attention to pertinent mathematical theory and its applications.

Volumes 34–35 (1987), 6 issues **ISSN 0022-0000**

In the U.S.A. and Canada: $270.00 **All other countries: $302.00**

For a free sample copy or for more information, please call:

ACADEMIC PRESS, INC., Journal Promotion Dept., 1250 Sixth Avenue, San Diego, CA 92101, U.S.A. (619) 230-1840

Please enter my subscription for:

☐ **Journal of Complexity,** Vol. 3 (1987) $_____

☐ **Journal of Parallel and Distributed Computing,** Vol. 4 (1987) $_____

☐ **Information and Computation,,** Vols. 72–75 (1987) $_____

☐ **Journal of Computer and System Sciences,** Vols. 34–35 (1987) $_____

Please send me:

☐ A sample copy of ____________________

☐ Privileged personal rate information. Personal subscription rates are available only on orders placed directly with the Publisher and paid for with personal funds. Checks and money orders must be drawn against a U.S. bank.

Please note: Payment or credit card information must accompany order. Prices include postage, handling, and air freight where applicable, and are subject to change without notice. Payment must be in U.S. currency, U.S. Bank Draft, International Money Order, or UNESCO coupons.

Method of payment:

☐ Our purchase order is enclosed.

☐ My check is enclosed. ☐ Credit information is given.

☐ Enter as a Standing Order (institutions only) hereby authorizing Academic Press to service and bill our account each calendar year until cancelled.

When paying by credit card

Please check one box: ☐ Visa/Barclaycard ☐ American Express ☐ MasterCard/Access ☐ Diner's Club

Expiration Date (Mo/Yr): __________

Credit card number: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Signature: ____________________

Total Amount: __________

Name: ____________________

Address: ____________________

City: ____________________

State: __________ Zip: __________

Country: ____________________

S7069

ADVERTISER INDEX

Academic Press, Inc.	119
Amdahl Corporation	114
AT&T Bell Laboratories	8
BDM International, Inc.	Cover II
CACI	1
CAIA '88	Cover III
Commerce Tours International, Inc.	96
Compsac '87	6
Computer Society Membership	88
Computer Society Ombudsman	59
Computer Standards Conference	81
CVPR '88	41
Data Communications Symposium	99
Digital Equipment Corporation	49
Hypertext '87	12
ICCD '87	4-5
ICDCS '87	28-29
Inmac	37
Interactive Development Environments	62
John Wiley & Sons Ltd.	94
Lockheed Missiles & Space Company	113
Rapitech Systems Inc.	13
Siggraph '87	Cover IV
Simulation of Computer Networks Symposium	61
Software Consulting Specialists, Inc.	73
Software Maintenance '87 Conference	74-75
Ultraware Ltd.	73
Wang Laboratories, Inc.	91

FOR DISPLAY ADVERTISING INFORMATION CONTACT

Northern California and Pacific Northwest: Don Farris Co., 161 W. 25th Ave., Ste. 102B, San Mateo, CA 94403; (415) 349-2222.
Southern California and Mountain States: Richard C. Faust Co., 24050 Madison St., Ste. 100, Torrance, CA 90505; (213) 373-9604.
Southwest: The House Co., 3000 Wesleyan, Houston, TX 77027; (713) 622-2868.
Midwest: Knorr Micromedia, Inc., 333 N. Michigan Ave., Chicago, IL 60601; (312) 726-2633.
East Coast: Hart Associates, PO Box 339, 42 Lake Blvd., Matawan, NJ 07747; (201) 583-8500.
New England: Arpin Associates, PO Box 227, Weston, MA 02193; (617) 899-5613.
Southeast/Telemarketing: Kay Young & Assoc., 22 Pebblewood, Irvine, CA 92714; (714) 551-4924.
Europe: Heinz J. Gorgens, Parkstrasse 8a, D-4054 Nettetal 1 - Hinsbeck (F.R.G).

Advertising Director: Dawn Peck.
For production information, conference, and classified advertising, contact Heidi Rex or Marian Tibayan.

COMPUTER, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 821-8380.

PRODUCT INDEX

	RS #	Page#
Boards		
Card	39,85	101,107
Extender	66	105
Prototyping	64,65	105
Circuits		
Microsystem Announ.	120-131	109
Modem	57,71,77,78	103,106
Conferences & Courses		
CAIA'88	—	Cover III
Compsac'87	—	6
Computer Networks	—	61
Computer Standards	—	81
CVPR'88	—	41
Data Communications	—	99
Distributed Computing Systems	—	28-29
Hypertext'87	—	12
ICCD'87	—	4-5
Siggraph'87	—	Cover IV
Simulation Conference 11	—	1
Software Maintenance	—	74-75
Data Acquisition		
Retrieval	59-61,75	104,106
Data Communications		
Gateway	62	104
Media-converter	69	105
Telecommunications program	51	102
I/O Related Equipment		
Diskette handler	70	105
Hand-held scanner	76	106
Other Products & Services		
Far East Tours	7	96
Publishers		
Book	6	94,119
Catalog	2	37
CS Membership	—	88
CS Ombudsman	—	59
Recruitment		
Technical professional	5	Cover II, 8,49,73,91,113, 116
Software		
Ada tool/training	86,87	107
AI	46	101
Data dictionary	4	73
Data management	40	101
G-Logis	47	102
Language converter	1	13
Security	41	101
Software tool	49,83	102,107
Systems		
Computer	36-38,67, 72-74,84	101,105, 106,107
Integrated system	35	100
Management	50,68,88	102,105, 107
Minisupercomputer	33,34	100
Multiuser	30-32	100
Network	58,63, 79-82	104,107
Operating system	53-56	103
Tool environment	3	62
Workstation	42-45,48,52	101,102

READER SERVICE CARD

COMPUTER

June 1987 issue

(Card void after November 1987)

Name________________

Title ________________

Company ________________

Address________________

City________ State ___ ZIP______

Country________ Phone (___) ______

Please send

(Circle those you want):

201 Publications catalog
202 Membership information
203 Student membership information
204 Application for senior membership

Reader interest

(Circle what you liked,
add comments on the back)

A Guest Editor's Introduction
B Fault-Tolerant Networks
C Shuffle-Exchange Networks
D Multiple Bus Architectures
E Optical Crossbar Networks
F Multicomputer Networks

J Open Channel
K Standards
L Update
M Computer Society News
N Conferences
P Calendar
Q Call for Papers
R New Products
S New Literature
T Microsystem Announcements
U IC Announcements
V Book Reviews

Product inquiries

1

(circle the numbers for products and advertisers
you want more information on)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	190
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

READER SERVICE CARD

COMPUTER

June 1987 issue

(Card void after November 1987)

Name________________

Title ________________

Company ________________

Address________________

City________ State ___ ZIP______

Country________ Phone (___) ______

Please send

(Circle those you want):

201 Publications catalog
202 Membership information
203 Student membership information
204 Application for senior membership

Reader interest

(Circle what you liked,
add comments on the back)

A Guest Editor's Introduction
B Fault-Tolerant Networks
C Shuffle-Exchange Networks
D Multiple Bus Architectures
E Optical Crossbar Networks
F Multicomputer Networks

J Open Channel
K Standards
L Update
M Computer Society News
N Conferences
P Calendar
Q Call for Papers
R New Products
S New Literature
T Microsystem Announcements
U IC Announcements
V Book Reviews

Product inquiries

2

(circle the numbers for products and advertisers
you want more information on)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	190
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

READER SERVICE CARD

COMPUTER

June 1987 issue

(Card void after November 1987)

Name________________

Title ________________

Company ________________

Address________________

City________ State ___ ZIP______

Country________ Phone (___) ______

Please send

(Circle those you want):

201 Publications catalog
202 Membership information
203 Student membership information
204 Application for senior membership

Reader interest

(Circle what you liked,
add comments on the back)

A Guest Editor's Introduction
B Fault-Tolerant Networks
C Shuffle-Exchange Networks
D Multiple Bus Architectures
E Optical Crossbar Networks
F Multicomputer Networks

J Open Channel
K Standards
L Update
M Computer Society News
N Conferences
P Calendar
Q Call for Papers
R New Products
S New Literature
T Microsystem Announcements
U IC Announcements
V Book Reviews

Product inquiries

3

(circle the numbers for products and advertisers
you want more information on)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	190
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

Editorial comments

I liked:

I disliked:

I would like to see:

For reader service inquiries, use other side

PO box address for
reader service cards only

PLACE
STAMP
HERE

COMPUTER
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91605
USA

Editorial comments

I liked:

I disliked:

I would like to see:

For reader service inquiries, use other side

PO box address for
reader service cards only

PLACE
STAMP
HERE

COMPUTER
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91605
USA

Editorial comments

I liked:

I disliked:

I would like to see:

For reader service inquiries, use other side

PO box address for
reader service cards only

PLACE
STAMP
HERE

COMPUTER
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91605
USA

CALL FOR PARTICIPATION

The Fourth IEEE Conference on ARTIFICIAL INTELLIGENCE APPLICATIONS

Sheraton Harbor Island Hotel
San Diego, California
March 14-18, 1988

sponsored by: The Computer Society of the IEEE

This conference is devoted to the application of artificial intelligence techniques to real-world problems. Only new, significant and previously unpublished work will be accepted. Two kinds of submissions are appropriate:

Those that focus on AI techniques or implementations and show how they can be applied effectively to important problems; and

Those that focus on particular AI-based application programs that solve significant problems, including an analysis of why the implementation techniques selected are appropriate for the problem domain.

AI techniques include: Knowledge representation, reasoning, uncertainty, knowledge acquisition, learning, natural language, intelligent interfaces, and general tools.

Application areas include: Science, medicine, law, business, engineering, manufacturing, and robotics

The following types of submissions will be reviewed by the Program Committee:

Full-length papers: 5000 words maximum, describing significant completed research.

Poster session abstracts: 1000 words maximum, describing interesting ongoing research.

Demonstration proposals: Videotape and/or description of a live presentation.

Panel discussion proposals: Topic and desired participants.

Submission information

Send all submissions to: Elaine Kant, Dennis O'Neill, Program Chairs, CAIA-88
Schlumberger-Doll Research, Old Quarry Road, Ridgefield, CT 06877-4108

Full-length papers: Submit four copies of the paper (5000 word limit) by September 18, 1987. The first page of the paper should contain the contact information for the author(s) (name, affiliation, address), a 200 word abstract, and a list of appropriate subject categories. Authors are not restricted to only those categories listed above. Accepted papers will be allocated six manuscript pages in the proceedings.

Poster session abstracts: Submit four copies of the abstract (1000 word limit) by November 25, 1987. Indicate all appropriate subject categories. Accepted abstracts will be reprinted and distributed at the conference. In addition, authors of accepted poster session abstracts will be provided with space at the conference to display examples of their work and to discuss their findings with others.

Demonstration proposals: Submit a videotape or a written description of a 10 to 15 minute demonstration by November 25, 1987. The demonstration should be of a particular system or technique that shows the reduction to practice of one of the conference topics.

Panel discussion proposals: Submit a one page description of a topic for panel discussion by November 25, 1987. Indicate the kind of speakers you would like to see participate and whether you are interested in organizing the discussion.

Important dates

Full-length papers due: September 18, 1987
Author notifications mailed: October 16, 1987
Other submissions due: November 25, 1987
Accepted papers to IEEE: December 4, 1987
Conference: March 14-18, 1988, San Diego, CA

Conference committee

General chair
James Miller, MCC
Tutorial chair:
Paul Harmon, Harmon Associates
Program committee chairs:
Elaine Kant, Schlumberger-Doll Research
Dennis O'Neill, Schlumberger-Doll Research
Program committee:
Richard Duda, Syntelligence
Mark Fox, Carnegie-Mellon University
Se June Hong, IBM T J. Watson Laboratories
Janet Kolodner, Georgia Institute of Technology
Casimir Kulikowski, Rutgers University
John Landry, Distribution Management Systems
Frank Lynch, Digital Equipment Corporation
Nancy Martin, Softpert Systems
Shirley McCarty, The Aerospace Corporation
Ryszard Michalski, University of Illinois
Sanjay Mittal, Xerox Palo Alto Research Center
Fumio Mizoguchi, Tokyo University of Science
Makoto Nagao, Kyoto University
Howard Shrobe, Symbolics
Marilyn Stelzner, Intellicorp
William Swartout, USC-ISI
Richard L Wexelblat, Philips Laboratories

Additional Information

For additional conference information, contact:
CAIA-88
The Computer Society of the IEEE
1730 Massachusetts Avenue, NW
Washington, DC 20036-1903
(202) 371-0101

For information about exhibits, contact:
Richard Greene
Trade Associates, Inc.
12250 Rockville Pike, Suite 200
Rockville, MD 20852
(301) 468-3210

THE COMPUTER SOCIETY OF THE IEEE

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS INC
IEEE

COMPUTER GRAPHICS THAT JOLT YOUR SENSES AND BRING YOUR IMAGINATION TO LIFE

ACM SIGGRAPH '87

1

2

14th Annual Conference
on Computer Graphics
and Interactive Techniques

July 27-31, 1987
Anaheim, California
Anaheim Convention Center

Sponsored by the Association for Computing Machinery's Special Interest Group on Computer Graphics in cooperation with the IEEE Technical Committee on Computer Graphics.

3

CREDITS
1. "Domo Domo Domo 1," Chisako ITO, Toyo Links Corporation, © 1987
2. "Gehause," Roman Kuchkuda, Megatek Corporation, © 1987.
3. "Grand Canyon," Digital Scene Simulation (sm) Omnibus Simulation, © 1987.

Please send a copy of the SIGGRAPH '87 Advance Program to:

NAME

COMPANY

ADDRESS

CITY STATE/COUNTRY

ZIP CODE TELEPHONE (   )

Return form to:
SIGGRAPH '87
Conference Management
111 East Wacker Drive, #600
Chicago, IL 60601 USA
(312) 644-6610

CS5

Members of ACM SIGGRAPH will automatically receive the advance program in April, and **NEED NOT** return this coupon.